

Romerito Campos de Andrade

Roteamento *Multicast* Multisessão -
Modelos e Algoritmos.

Natal, Brasil

2018

Romerito Campos de Andrade

Roteamento *Multicast* Multisessão - Modelos e Algoritmos.

Tese de doutorado apresentada ao programa de Pós-Graduação em Sistema e Computação do Departamento de Informática e Matemática aplicada como pré-requisito para obtenção do grau de Doutor em Ciência da Computação.

Universidade Federal do Rio Grande do Norte
Departamento de Matemática e Informática Aplicada
Programa de Pós-Graduação em Sistemas e Computação

Orientador: Marco César Goldbarg

Natal, Brasil
2018

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Setorial Prof. Ronaldo Xavier de Arruda - CCET

Andrade, Romerito Campos de.

Roteamento multicast multisessão: modelos e algoritmos /
Romerito Campos de Andrade. - 2018.
329f.: il.

Tese (doutorado) - Universidade Federal do Rio Grande do
Norte, Centro de Ciências Exatas e da Terra, Programa de Pós-
Graduação em Sistemas e Computação. Natal, 2018.

Orientador: Marco César Goldbarg.

1. Computação - Tese. 2. Multicast - Tese. 3. Multiobjetivo -
Tese. 4. Multi-fonte - Tese. 5. Multisessão - Tese. 6. Diagramas
de Voronoi - Tese. 7. Transgenético - Tese. I. Goldbarg, Marco
César. II. Título.

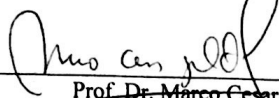
RN/UF/CCET

CDU 004

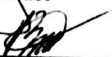
ROMERITO CAMPOS DE ANDRADE

Roteamento Multicast Multisessão - Modelos e Algoritmos

Esta Tese foi julgada adequada para a obtenção do título de doutor em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte.

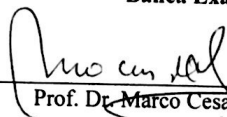


Prof. Dr. Marco Cesar Goldbarg – UFRN
Orientador

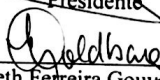


Prof. Dr. Bruno Motta de Carvalho – UFRN
Coordenador do Programa

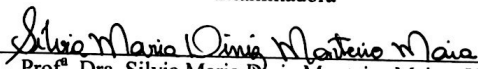
Banca Examinadora



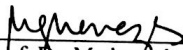
Prof. Dr. Marco Cesar Goldbarg – UFRN
Presidente



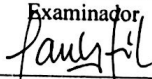
Prof.ª Dra. Elizabeth Ferreira Gouveia Goldbarg – UFRN
Examinadora



Prof.ª Dra. Silvia Maria Diniz Monteiro Maia – UFRN
Examinadora



Prof. Dr. Matheus da Silva Menezes – UFRSA
Examinador



Prof. Dr. Paulo Henrique Asconavieta da Silva – IFRS
Examinador

Maio, 2018

Dedico este trabalho aos meus pais e a minha esposa Márcia Rocha.

Agradecimentos

Em primeiro lugar, à Deus.

Aos amigos mais próximo que me ajudaram em diversos momentos.

Aqueles que mantêm o DIMAP em pleno funcionamento permitindo mais tranquilidade para o desenvolvimento das atividades.

Aos participantes do LAE, principalmente aqueles que estão comigo no dia-dia do laboratório.

Aos Professores Marco e Elizabeth pelo empenho na orientação do trabalho.

“A riqueza é comum, mas o saber é raro.

Dan Brown

Resumo

A tecnologia *multicast* tem sido amplamente estudada ao longo dos anos e apresenta-se como uma solução para melhor utilização dos recursos da rede. Várias abordagens já foram avaliadas para o problema de roteamento desde o uso de uma sessão com apenas uma fonte a um cenário com múltiplas sessões e múltiplas fontes por sessão. Neste trabalho, é feito um estudo dos modelos matemáticos para o problema com múltiplas sessões e múltiplas fontes. Dois modelos matemáticos foram propostos: uma versão multissessão mono-objetivo que visa a otimização da capacidade residual sujeito a um limite de custo e uma versão multiobjetivo com três funções-objetivo. Ambos os modelos levam em conta o cenário multissessão com uma fonte por sessão. Além disso, um estudo algorítmico foi realizado sobre um modelo da literatura que utiliza múltiplas fontes por sessão. Três conjuntos de algoritmos foram propostos. O primeiro conjunto trata do problema mono-objetivo proposto e considera as abordagens ACO, Genético, GRASP e ILS. O segundo conjunto consiste dos algoritmos propostos para o modelo multiobjetivo. Foram projetados os seguintes algoritmos: NSGA2, ssNSGA2, GDE3, MOEA/D e SMS-EMOA. Além disso, foi projetado um algoritmo transgenético com subpopulações baseadas em operadores de criação de solução direcionados por objetivos do problema. Também foi utilizado o conceito de soluções de elite. No total, 8 versões do algoritmo transgenético foram avaliadas. O terceiro conjunto de algoritmos consiste da heurística MMVD proposta para o modelo da literatura com múltiplas fontes por sessão. Esta heurística é baseada no uso de diagramas de Voronoi. O processo experimental foi realizado com amplo número de instâncias configuradas de modo a avaliar diferentes situações. Os resultados foram comparados utilizando métodos estatísticos não-paramétricos. A análise final indicou que o ILS e o Genético obtiveram resultados muito similares, entretanto o ILS possui melhor tempo de processamento. A versão **cross0** do algoritmo transgenético obteve o melhor resultado em praticamente todos os cenários avaliados. A heurística MMVD obteve excelentes resultados sobre algoritmos da literatura.

Palavras-chave: *Multicast*. Multiobjetivo. Multi-Fonte. Multissessão. Transgenético. Diagramas de Voronoi.

Abstract

Multicast Technology has been studied over the last two decades and It has shown to be a good approach to save network resources. Many approaches have been considered to solve the multicast routing problem considering only one session and one source to attending session's demand, as well, multiple sessions with more than one source per session. In this thesis, the multicast routing problem is explored taking in consideration the models and the algorithms designed to solve it when where multiple sessions and sources. Two new models are proposed with different focuses. First, a mono-objective model optimizing residual capacity, Z , of the network subject to a budget is designed and the objective is to maximize Z . Second, a multi-objective model is designed with three objective functions: cost, Z and hops counting. Both models consider multisession scenario with one source per session. Besides, a third model is examined. This model was designed to optimize Z in a scenario with multiple sessions with support to more than one source per session. An experimental analysis was realized over the models considered. For each model, a set of algorithms were designed. First, an ACO, a Genetic algorithm, a GRASP and an ILS algorithm were designed to solve the mono-objective model – optimizing Z subject to a budget. Second, a set of algorithm were designed to solve the multi-objective model. The classical approaches were used: NSGA2, ssNSGA2, SMS-EMOA, GDE3 and MOEA/D. In addition, a transgenetic algorithm was designed to solve the problem and it was compared against the classical approaches. This algorithm considers the use of subpopulations during the evolution. Each subpopulation is based on a solution construction operator guided by one of the objective functions. Some solutions are considered as elite solutions and they are considered to be improved by a transposon operator. Eight versions of the transgenetic algorithm were evaluated. Third, an algorithm was designed to solve the problem with multiple sessions and multiple sources per sessions. This algorithm is based on Voronoi Diagrams and it is called MMVD. The algorithm designed were evaluated on large experimental analysis. The sample generated by each algorithm on the instances were evaluated based on non-parametric statistical tests. The analysis performed indicates that ILS and Genetic algorithm have outperformed the ACO and GRASP. The comparison between ILS and Genetic has shown that ILS has better processing time performance. In the multi-objective scenario, the version of Transgenetic called cross0 has shown to be statistically better than the other algorithms in most of the instances based on the hypervolume and additive/multiplicative epsilon quality indicators. Finally, the MMVD algorithm has shown to be better than the algorithm from literature based on the experimental analysis performed for the model with multiple session and multiple sources per session.

Keywords: *multicast*. Multiobjective. Multi-Source. Multi-Session. Transgenetic. Voronoi Diagrams.

Lista de ilustrações

Figura 1 – Exemplo de fluxo de dados em uma rede IP.	46
Figura 2 – Exemplo de fluxo de dados <i>broadcast</i>	46
Figura 3 – Exemplo de fluxo de dados em uma rede com IP <i>multicast</i>	47
Figura 4 – Associação entre endereço IP <i>multicast</i> e endereço MAC.	48
Figura 5 – Árvore de distribuição <i>Multicast</i>	49
Figura 6 – Transmissão em Árvore Compartilhada.	50
Figura 7 – Funcionamento do Algoritmo RPF.	51
Figura 8 – Comunicação entre AS's usando BGP.	54
Figura 9 – Abordagens de construção de árvores de distribuição e protocolos. . . .	55
Figura 10 – Exemplo gráfico de hipervolume de conjunto de aproximação.	82
Figura 11 – Exemplos de conjuntos de aproximação para avaliação com indicador I_{ϵ} . . .	83
Figura 12 – Classificação da população em fronteiras de não-dominados.	85
Figura 13 – Exemplo de região que determina a distância de <i>crowding</i> do indivíduo j . . .	88
Figura 14 – Exemplo de uso do critério de seleção baseado em hipervolume e número de indivíduos dominantes.	92
Figura 15 – Mutação utilizada no algoritmo DE.	94
Figura 16 – Crossover utilizado no algoritmo DE.	94
Figura 17 – Processo de substituição de arestas por criação-remoção de ciclos. . . .	112
Figura 18 – Representação de uma solução: estrutura utilizada (cromossomo) e árvore associada a representação.	120
Figura 19 – Exemplo de criação de diagrama de Voronoi em grafos.	128
Figura 20 – Exemplo do procedimento consolidar árvore.	136
Figura 21 – Fluxo de Execução do algoritmo transgenético sem uso de cruzamento. . .	140
Figura 22 – Fluxo de Execução do algoritmo transgenético com cruzamento de so- luções.	142
Figura 23 – Avaliação de escalabilidade com relação a tempo de processamento. . .	160
Figura 24 – Ilustração de resultados por meio de Mapa de calor.	165
Figura 25 – Resultado do PostHoc de Nemenyi sobre os resultado da tabela 23 para o I_H na instância B30_01.	175
Figura 26 – Comparação dos algoritmos por Capacidade Residual no conjunto $n =$ 30 por valor de W . Os resultados correspondem a média de 30 execu- ções em cada instância.	188
Figura 27 – Comparação dos algoritmos por Tempo de Processamento no conjunto com $n = 30$ por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.	189

Figura 28 – Comparação dos algoritmos por Capacidade Residual no conjunto $n = 60$ por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.	191
Figura 29 – Comparação dos algoritmos por Tempo de Processamento no conjunto com $n = 60$ e por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.	192
Figura 30 – Comparação dos algoritmos por Capacidade Residual no conjunto com $n = 120$ por valor de W . Os resultados corresponde a média de 30 execuções em cada instância.	194
Figura 31 – Comparação dos algoritmos por Tempo de Processamento no conjunto com $n = 120$ e por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.	195
Figura 32 – Comparação dos algoritmos por Capacidade Residual com $n = 240$ e por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.	197
Figura 33 – Comparação dos algoritmos por Tempo de Processamento com $n = 240$ e por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.	198
Figura 34 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 20$	212
Figura 35 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $n = 20$	213
Figura 36 – <i>Posthoc</i> de Shaffer para I_{ϵ} no conjunto com $n = 20$	214
Figura 37 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $n = 20$	215
Figura 38 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 40$	216
Figura 39 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $n = 40$	217
Figura 40 – <i>Posthoc</i> de Shaffer para I_{ϵ} no conjunto com $n = 40$	218
Figura 41 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $n = 40$	219
Figura 42 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 60$	220
Figura 43 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $n = 60$	221
Figura 44 – <i>Posthoc</i> de Shaffer para I_{ϵ} no conjunto com $n = 60$	222
Figura 45 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $n = 60$	223
Figura 46 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 80$	224
Figura 47 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $n = 80$	225
Figura 48 – <i>Posthoc</i> de Shaffer para I_{ϵ} no conjunto com $n = 80$	226
Figura 49 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $n = 80$	227
Figura 50 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 100$	228
Figura 51 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $n = 100$	229
Figura 52 – <i>Posthoc</i> de Shaffer para I_{ϵ} no conjunto com $n = 100$	230
Figura 53 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $n = 100$	231
Figura 54 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $D^k=10$	232

Figura 55 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $D^k = 10$	233
Figura 56 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $D^k = 10$	234
Figura 57 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $D^k = 10$	235
Figura 58 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $D^k = 20$	236
Figura 59 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $D^k = 20$	237
Figura 60 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $D^k = 20$	238
Figura 61 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $D^k = 20$	239
Figura 62 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $D^k = 30$	240
Figura 63 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $D^k = 30$	241
Figura 64 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $D^k = 30$	242
Figura 65 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $D^k = 30$	243
Figura 66 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $D^k = 40$	244
Figura 67 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $D^k = 40$	245
Figura 68 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $D^k = 40$	246
Figura 69 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $D^k = 40$	247
Figura 70 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $D^k = 50$	248
Figura 71 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $D^k = 50$	249
Figura 72 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $D^k = 50$	250
Figura 73 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $D^k = 50$	251
Figura 74 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $W = 10$	252
Figura 75 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $W = 10$	253
Figura 76 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $W = 10$	254
Figura 77 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $W = 10$	255
Figura 78 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $W = 20$	256
Figura 79 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $W = 20$	257
Figura 80 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $W = 20$	258
Figura 81 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $W = 20$	259
Figura 82 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $W = 30$	260
Figura 83 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $W = 30$	261
Figura 84 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $W = 30$	262
Figura 85 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $W = 30$	263
Figura 86 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $W = 40$	264
Figura 87 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $W = 40$	265
Figura 88 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $W = 40$	266
Figura 89 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $W = 40$	267
Figura 90 – <i>Posthoc</i> de Shaffer para $I_{\epsilon+}$ no conjunto com $W = 50$	268
Figura 91 – <i>Posthoc</i> de Shaffer para I_H no conjunto com $W = 50$	269
Figura 92 – <i>Posthoc</i> de Shaffer para I_ϵ no conjunto com $W = 50$	270
Figura 93 – <i>Posthoc</i> de Shaffer para o Tempo no conjunto com $W = 50$	271

Figura 94 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_01 e $W = 5$	281
Figura 95 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_01 e $W = 5$	281
Figura 96 – <i>Posthoc</i> de Nemenyi I_H para b30_01 e $W = 5$	281
Figura 97 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_01 e $W = 5$	281
Figura 98 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_02 e $W = 5$	282
Figura 99 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_02 e $W = 5$	282
Figura 100 – <i>Posthoc</i> de Nemenyi I_H para b30_02 e $W = 5$	282
Figura 101 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_02 e $W = 5$	282
Figura 102 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_04 e $W = 5$	283
Figura 103 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_04 e $W = 5$	283
Figura 104 – <i>Posthoc</i> de Nemenyi I_H para b30_04 e $W = 5$	283
Figura 105 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_04 e $W = 5$	283
Figura 106 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_07 e $W = 10$	284
Figura 107 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_07 e $W = 10$	284
Figura 108 – <i>Posthoc</i> de Nemenyi I_H para b30_07 e $W = 10$	284
Figura 109 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_07 e $W = 10$	284
Figura 110 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_08 e $W = 10$	285
Figura 111 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_08 e $W = 10$	285
Figura 112 – <i>Posthoc</i> de Nemenyi I_H para b30_08 e $W = 10$	285
Figura 113 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_08 e $W = 10$	285
Figura 114 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_09 e $W = 10$	286
Figura 115 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_09 e $W = 10$	286
Figura 116 – <i>Posthoc</i> de Nemenyi I_H para b30_09 e $W = 10$	286
Figura 117 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_09 e $W = 10$	286
Figura 118 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_13 e $W = 15$	287
Figura 119 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_13 e $W = 15$	287
Figura 120 – <i>Posthoc</i> de Nemenyi I_H para b30_13 e $W = 15$	287
Figura 121 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_13 e $W = 15$	287
Figura 122 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_14 e $W = 15$	288
Figura 123 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_14 e $W = 15$	288
Figura 124 – <i>Posthoc</i> de Nemenyi I_H para b30_14 e $W = 15$	288
Figura 125 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_14 e $W = 15$	288
Figura 126 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_15 e $W = 15$	289
Figura 127 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_15 e $W = 15$	289
Figura 128 – <i>Posthoc</i> de Nemenyi I_H para b30_15 e $W = 15$	289
Figura 129 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_15 e $W = 15$	289
Figura 130 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b30_19 e $W = 20$	290
Figura 131 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_19 e $W = 20$	290
Figura 132 – <i>Posthoc</i> de Nemenyi I_H para b30_19 e $W = 20$	290

Figura 133 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_19 e $W = 20$	290
Figura 134 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b30_20 e $W = 20$	291
Figura 135 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_20 e $W = 20$	291
Figura 136 – <i>Posthoc</i> de Nemenyi I_H para b30_20 e $W = 20$	291
Figura 137 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_20 e $W = 20$	291
Figura 138 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b30_21 e $W = 20$	292
Figura 139 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_21 e $W = 20$	292
Figura 140 – <i>Posthoc</i> de Nemenyi I_H para b30_21 e $W = 20$	292
Figura 141 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_21 e $W = 20$	292
Figura 142 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b30_25 e $W = 25$	293
Figura 143 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_25 e $W = 25$	293
Figura 144 – <i>Posthoc</i> de Nemenyi I_H para b30_25 e $W = 25$	293
Figura 145 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_25 e $W = 25$	293
Figura 146 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b30_26 e $W = 25$	294
Figura 147 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_26 e $W = 25$	294
Figura 148 – <i>Posthoc</i> de Nemenyi I_H para b30_26 e $W = 25$	294
Figura 149 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_26 e $W = 25$	294
Figura 150 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b30_27 e $W = 25$	295
Figura 151 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b30_27 e $W = 25$	295
Figura 152 – <i>Posthoc</i> de Nemenyi I_H para b30_27 e $W = 25$	295
Figura 153 – <i>Posthoc</i> de Nemenyi Tempo(s) para b30_27 e $W = 25$	295
Figura 154 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b60_01 e $W = 5$	296
Figura 155 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_01 e $W = 5$	296
Figura 156 – <i>Posthoc</i> de Nemenyi I_H para b60_01 e $W = 5$	296
Figura 157 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_01 e $W = 5$	296
Figura 158 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b60_02 e $W = 5$	297
Figura 159 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_02 e $W = 5$	297
Figura 160 – <i>Posthoc</i> de Nemenyi I_H para b60_02 e $W = 5$	297
Figura 161 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_02 e $W = 5$	297
Figura 162 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b60_03 e $W = 5$	298
Figura 163 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_03 e $W = 5$	298
Figura 164 – <i>Posthoc</i> de Nemenyi I_H para b60_03 e $W = 5$	298
Figura 165 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_03 e $W = 5$	298
Figura 166 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b60_07 e $W = 10$	299
Figura 167 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_07 e $W = 10$	299
Figura 168 – <i>Posthoc</i> de Nemenyi I_H para b60_07 e $W = 10$	299
Figura 169 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_07 e $W = 10$	299
Figura 170 – <i>Posthoc</i> de Nemenyi $I_{\epsilon+}$ para b60_08 e $W = 10$	300
Figura 171 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_08 e $W = 10$	300

Figura 172 – <i>Posthoc</i> de Nemenyi I_H para b60_08 e $W = 10$	300
Figura 173 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_08 e $W = 10$	300
Figura 174 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_09 e $W = 10$	301
Figura 175 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_09 e $W = 10$	301
Figura 176 – <i>Posthoc</i> de Nemenyi I_H para b60_09 e $W = 10$	301
Figura 177 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_09 e $W = 10$	301
Figura 178 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_13 e $W = 15$	302
Figura 179 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_13 e $W = 15$	302
Figura 180 – <i>Posthoc</i> de Nemenyi I_H para b60_13 e $W = 15$	302
Figura 181 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_13 e $W = 15$	302
Figura 182 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_14 e $W = 15$	303
Figura 183 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_14 e $W = 15$	303
Figura 184 – <i>Posthoc</i> de Nemenyi I_H para b60_14 e $W = 15$	303
Figura 185 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_14 e $W = 15$	303
Figura 186 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_15 e $W = 15$	304
Figura 187 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_15 e $W = 15$	304
Figura 188 – <i>Posthoc</i> de Nemenyi I_H para b60_15 e $W = 15$	304
Figura 189 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_15 e $W = 15$	304
Figura 190 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_19 e $W = 20$	305
Figura 191 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_19 e $W = 20$	305
Figura 192 – <i>Posthoc</i> de Nemenyi I_H para b60_19 e $W = 20$	305
Figura 193 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_19 e $W = 20$	305
Figura 194 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_20 e $W = 20$	306
Figura 195 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_20 e $W = 20$	306
Figura 196 – <i>Posthoc</i> de Nemenyi I_H para b60_20 e $W = 20$	306
Figura 197 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_20 e $W = 20$	306
Figura 198 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_21 e $W = 20$	307
Figura 199 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_21 e $W = 20$	307
Figura 200 – <i>Posthoc</i> de Nemenyi I_H para b60_21 e $W = 20$	307
Figura 201 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_21 e $W = 20$	307
Figura 202 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_25 e $W = 25$	308
Figura 203 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_25 e $W = 25$	308
Figura 204 – <i>Posthoc</i> de Nemenyi I_H para b60_25 e $W = 25$	308
Figura 205 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_25 e $W = 25$	308
Figura 206 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_26 e $W = 25$	309
Figura 207 – <i>Posthoc</i> de Nemenyi I_{ϵ} para b60_26 e $W = 25$	309
Figura 208 – <i>Posthoc</i> de Nemenyi I_H para b60_26 e $W = 25$	309
Figura 209 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_26 e $W = 25$	309
Figura 210 – <i>Posthoc</i> de Nemenyi I_{ϵ_+} para b60_27 e $W = 25$	310

Figura 211 – <i>Posthoc</i> de Nemenyi I_ϵ para b60_27 e $W = 25$	310
Figura 212 – <i>Posthoc</i> de Nemenyi I_H para b60_27 e $W = 25$	310
Figura 213 – <i>Posthoc</i> de Nemenyi Tempo(s) para b60_27 e $W = 25$	310

Lista de tabelas

Tabela 1 – Principais demandas de QoS.	57
Tabela 2 – Tabela adaptada de (KNOWLES; THIELE; ZITZLER, 2006) ilustrando as relações entre vetores objetivos.	79
Tabela 3 – Extensão das relações de dominância aplicadas a conjuntos de aproximação.	80
Tabela 4 – Procedimentos utilizados nos vetores transgenéticos (GOLDBARG; GOLDBARG, 2002).	101
Tabela 5 – Vetores Transgenéticos (GOLDBARG; GOLDBARG, 2002).	102
Tabela 6 – Conjunto de Instâncias 1.	146
Tabela 7 – Conjunto de Instâncias 2.	148
Tabela 8 – Conjunto de Instâncias 3.	149
Tabela 9 – Resultado do Gurobi para $n = 30$ (esquerda) e $n = 60$ (direita).	151
Tabela 10 – Resultado do Gurobi para $n = 120$ (esquerda) e $n = 240$ (direita).	152
Tabela 11 – Parâmetros de entrada do IRACE.	153
Tabela 12 – Configurações de parâmetros utilizadas nos algoritmos.	153
Tabela 13 – Comparação de tempo de processamento <i>Solver</i> /Algoritmo proposto.	156
Tabela 14 – Valores utilizados como entrada para configuração de parâmetros pelo IRACE.	161
Tabela 15 – Parâmetros de entrada dos algoritmos obtidos através do IRACE.	162
Tabela 16 – Algoritmos e operadores utilizados nos experimentos.	163
Tabela 17 – Mediana de 30 execuções da cada algoritmo nas instâncias com $n = 20$. Cada linha representa uma instância contendo a mediana dos algoritmos com relação ao valor do indicador I_ϵ	164
Tabela 18 – Contagem de resultados em relação ao indicador I_ϵ no tamanho da rede (n).	166
Tabela 19 – Contagem de melhores resultados por algoritmo/indicador e número de vértices da rede (n).	169
Tabela 20 – Contagem de melhores resultados por algoritmo/indicador no número de participantes por sessão (D^k).	170
Tabela 21 – Contagem de melhores resultados por algoritmo/indicador de qualidade no tamanho da sessão (W).	171
Tabela 22 – Instâncias com Pareto ótimo utilizadas no experimento de comparação dos algoritmos multiobjetivo.	172
Tabela 23 – Hipervolume das execuções dos algoritmos na instância B30_01.	174
Tabela 24 – Resumo de contagem de vitórias e percentual total obtido por parâmetro W	178

Tabela 25 – Contagem de melhores resultado no $I_{\epsilon+}$.	179
Tabela 26 – Contagem de melhores resultado no I_{ϵ} .	179
Tabela 27 – Contagem de melhores resultado no I_H .	179
Tabela 28 – Contagem de melhores resultado no Tempo.	179
Tabela 29 – <i>Posthoc</i> de Nemenyi sobre resultado dos algoritmos por valor de W nas instâncias do conjunto com $\mathbf{n} = \mathbf{30}$ utilizando nível de significância igual 0,05.	190
Tabela 30 – <i>PostHoc</i> de Nemenyi sobre resultado dos algoritmos por valor de W nas instâncias do conjunto com $\mathbf{n} = \mathbf{60}$ e nível de significância igual a 0,05.	193
Tabela 31 – <i>PostHoc</i> de Nemenyi sobre resultado dos algoritmos por valor de W nas instâncias do conjunto com $n = 120$ e nível de significância igual a 0,05.	196
Tabela 32 – <i>PostHoc</i> de Nemenyi sobre resultado dos algoritmos por valor de W nas instâncias do conjunto 1 com $n = 240$ e nível de significância igual a 0,05.	199
Tabela 33 – <i>PostHoc</i> de Nemenyi aplicado aos resultados obtidos na comparação dos algoritmos diante da variação no tamanho da rede com $n \in \{30, 60, 120, 240\}$ e nível de significância igual a 0,05.	200
Tabela 34 – Mediana dos resultados obtidos no cenário 1. Cada valor de parâmetro possui 30 instâncias e a mediana corresponde ao resultado do algoritmo nesses instâncias.	202
Tabela 35 – Resultado do teste Post-hoc de Shaffer sobre as amostras obtidas pelos algoritmos no cenário 1 e parâmetros r, d e n . O nível de significância utilizado foi 0,05.	203
Tabela 36 – Mediana dos resultados obtidos no cenário 2. Cada valor de parâmetro possui 30 instâncias e a mediana corresponde ao resultado do algoritmo nesses instâncias.	204
Tabela 37 – Resultado do teste Post-hoc de Shaffer sobre as amostras obtidas pelos algoritmos no cenário 2 e parâmetros r, d e n . O nível de significância utilizado foi 0,05.	205
Tabela 38 – Mediana dos resultados obtidos no cenário 3. Cada valor de parâmetro possui 30 instâncias e a mediana corresponde ao resultado do algoritmo nesses instâncias.	206
Tabela 39 – Resultado do teste Post-hoc de Shaffer sobre as amostras obtidas pelos algoritmos no cenário 3 e parâmetros r, d, n e s . O nível de significância utilizado foi 0,05.	207

Tabela 40 – Mediana dos resultados obtidos no cenário 4. Cada valor de parâmetro possui 30 instâncias e a mediana corresponde ao resultado do algoritmo nessas instâncias.	208
Tabela 41 – Resultado do teste Post-hoc de Shaffer sobre as amostras obtidas pelos algoritmos no cenário 4 e parâmetros r, d, n e s . O nível de significância utilizado foi 0,05.	209
Tabela 42 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $n = 20$	212
Tabela 43 – Medianas do indicador de qualidade I_H no conjunto com $n = 20$	213
Tabela 44 – Medianas do indicador de qualidade I_{ϵ} no conjunto com $n = 20$	214
Tabela 45 – Medianas do tempo de processamento no conjunto com $n = 20$	215
Tabela 46 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $n = 40$	216
Tabela 47 – Medianas do indicador de qualidade I_H no conjunto com $n = 40$	217
Tabela 48 – Medianas do indicador de qualidade I_{ϵ} no conjunto com $n = 40$	218
Tabela 49 – Medianas do tempo de processamento no conjunto com $n = 40$	219
Tabela 50 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $n = 60$	220
Tabela 51 – Medianas do indicador de qualidade I_H no conjunto com $n = 60$	221
Tabela 52 – Medianas do indicador de qualidade I_{ϵ} no conjunto com $n = 60$	222
Tabela 53 – Medianas do tempo de processamento no conjunto com $n = 60$	223
Tabela 54 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $n = 80$	224
Tabela 55 – Medianas do indicador de qualidade I_H no conjunto com $n = 80$	225
Tabela 56 – Medianas do indicador de qualidade I_{ϵ} no conjunto com $n = 80$	226
Tabela 57 – Medianas do tempo de processamento no conjunto com $n = 80$	227
Tabela 58 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $n = 100$	228
Tabela 59 – Medianas do indicador de qualidade I_H no conjunto com $n = 100$	229
Tabela 60 – Medianas do indicador de qualidade I_{ϵ} no conjunto com $n = 100$	230
Tabela 61 – Medianas do tempo de processamento no conjunto com $n = 100$	231
Tabela 62 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $D^k = 10$	232
Tabela 63 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 10$	233
Tabela 64 – Medianas do indicador de qualidade I_{ϵ} no conjunto com $D^k = 10$	234
Tabela 65 – Medianas do tempo de processamento no conjunto com $D^k = 10$	235
Tabela 66 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $D^k = 20$	236
Tabela 67 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 20$	237
Tabela 68 – Medianas do indicador de qualidade I_{ϵ} no conjunto com $D^k = 20$	238
Tabela 69 – Medianas do tempo de processamento no conjunto com $D^k = 20$	239
Tabela 70 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $D^k = 30$	240
Tabela 71 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 30$	241
Tabela 72 – Medianas do indicador de qualidade I_{ϵ} no conjunto com $D^k = 30$	242
Tabela 73 – Medianas do tempo de processamento no conjunto com $D^k = 30$	243
Tabela 74 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $D^k = 40$	244

Tabela 75 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 40$	245
Tabela 76 – Medianas do indicador de qualidade I_ϵ no conjunto com $D^k = 40$	246
Tabela 77 – Medianas do tempo de processamento no conjunto com $D^k = 40$	247
Tabela 78 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $D^k = 50$	248
Tabela 79 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 50$	249
Tabela 80 – Medianas do indicador de qualidade I_ϵ no conjunto com $D^k = 50$	250
Tabela 81 – Medianas do tempo de processamento no conjunto com $D^k = 50$	251
Tabela 82 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $W = 10$	252
Tabela 83 – Medianas do indicador de qualidade I_H no conjunto com $W = 10$	253
Tabela 84 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 10$	254
Tabela 85 – Medianas do tempo de processamento no conjunto com $W = 10$	255
Tabela 86 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $W = 20$	256
Tabela 87 – Medianas do indicador de qualidade I_H no conjunto com $W = 20$	257
Tabela 88 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 20$	258
Tabela 89 – Medianas do tempo de processamento no conjunto com $W = 20$	259
Tabela 90 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $W = 30$	260
Tabela 91 – Medianas do indicador de qualidade I_H no conjunto com $W = 30$	261
Tabela 92 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 30$	262
Tabela 93 – Medianas do tempo de processamento no conjunto com $W = 30$	263
Tabela 94 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $W = 40$	264
Tabela 95 – Medianas do indicador de qualidade I_H no conjunto com $W = 40$	265
Tabela 96 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 40$	266
Tabela 97 – Medianas do tempo de processamento no conjunto com $W = 40$	267
Tabela 98 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $W = 50$	268
Tabela 99 – Medianas do indicador de qualidade I_H no conjunto com $W = 50$	269
Tabela 100 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 50$	270
Tabela 101 – Medianas do tempo de processamento no conjunto com $W = 50$	271
Tabela 102 – Contagem de resultados em relação ao indicador $I_{\epsilon+}$ no tamanho da rede(n).	272
Tabela 103 – Contagem de resultados em relação ao indicador I_H no tamanho da rede(n).	272
Tabela 104 – Contagem de resultados em relação ao indicador I_ϵ no tamanho da rede (n).	273
Tabela 105 – Contagem de resultados em relação ao Tempo de processamento no tamanho da rede (n)	273
Tabela 106 – Contagem de resultados em relação ao indicador $I_{\epsilon+}$ no número de participantes (D^k)	274
Tabela 107 – Contagem de resultados em relação ao indicador I_H no número de participantes (D^k).	274

Tabela 108–Contagem de resultados em relação ao indicador I_ϵ no número de participantes (D^k).	275
Tabela 109–Contagem de resultados em relação ao Tempo de processamento no número de participantes (D^k).	275
Tabela 110–Contagem de resultados em relação ao indicador $I_{\epsilon+}$ no número de sessões (W).	276
Tabela 111–Contagem de resultados em relação ao indicador I_H no número de sessões (W).	276
Tabela 112–Contagem de resultados em relação ao indicador I_ϵ no número de sessões (W).	277
Tabela 113–Contagem de resultados em relação ao Tempo de processamento no número de sessões (W).	277
Tabela 114–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=5$	279
Tabela 115–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=5$	280
Tabela 116–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=5$	311
Tabela 117–Contagem de melhores resultado no I_ϵ e $W=5$	311
Tabela 118–Contagem de melhores resultado no I_H e $W=5$	311
Tabela 119–Contagem de melhores resultado no Tempo e $W=5$	311
Tabela 120–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=5$	312
Tabela 121–Contagem de melhores resultado no I_ϵ e $W=5$	312
Tabela 122–Contagem de melhores resultado no I_H e $W=5$	312
Tabela 123–Contagem de melhores resultado no Tempo e $W=5$	312
Tabela 124–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=10$	313
Tabela 125–Contagem de melhores resultado no I_ϵ e $W=10$	313
Tabela 126–Contagem de melhores resultado no I_H e $W=10$	313
Tabela 127–Contagem de melhores resultado no Tempo e $W=10$	313
Tabela 128–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=10$	314
Tabela 129–Contagem de melhores resultado no I_ϵ e $W=10$	314
Tabela 130–Contagem de melhores resultado no I_H e $W=10$	314
Tabela 131–Contagem de melhores resultado no Tempo e $W=10$	314
Tabela 132–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=15$	315
Tabela 133–Contagem de melhores resultado no I_ϵ e $W=15$	315
Tabela 134–Contagem de melhores resultado no I_H e $W=15$	315
Tabela 135–Contagem de melhores resultado no Tempo e $W=15$	315
Tabela 136–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=15$	316
Tabela 137–Contagem de melhores resultado no I_ϵ e $W=15$	316
Tabela 138–Contagem de melhores resultado no I_H e $W=15$	316
Tabela 139–Contagem de melhores resultado no Tempo e $W=15$	316
Tabela 140–Contagem de melhores resultado no $I_{\epsilon+}$ e $W=20$	317

Tabela 141 – Contagem de melhores resultado no I_ϵ e $W=20$	317
Tabela 142 – Contagem de melhores resultado no I_H e $W=20$	317
Tabela 143 – Contagem de melhores resultado no Tempo e $W=20$	317
Tabela 144 – Contagem de melhores resultado no $I_{\epsilon+}$ e $W=20$	318
Tabela 145 – Contagem de melhores resultado no I_ϵ e $W=20$	318
Tabela 146 – Contagem de melhores resultado no I_H e $W=20$	318
Tabela 147 – Contagem de melhores resultado no Tempo e $W=20$	318
Tabela 148 – Contagem de melhores resultado no $I_{\epsilon+}$ e $W=25$	319
Tabela 149 – Contagem de melhores resultado no I_ϵ e $W=25$	319
Tabela 150 – Contagem de melhores resultado no I_H e $W=25$	319
Tabela 151 – Contagem de melhores resultado no Tempo e $W=25$	319
Tabela 152 – Contagem de melhores resultado no $I_{\epsilon+}$ e $W=25$	320
Tabela 153 – Contagem de melhores resultado no I_ϵ e $W=25$	320
Tabela 154 – Contagem de melhores resultado no I_H e $W=25$	320
Tabela 155 – Contagem de melhores resultado no Tempo e $W=25$	320

Lista de Algoritmos

1	Algoritmo para classificação da população (DEB et al., 2002).	86
2	Algoritmo para cálculo da distância de <i>crowding</i>	88
3	Non-Dominating Sorting Genetic Algorithm - NSGA2.	90
4	Algoritmo para otimização de Pareto com seleção baseada em hipervolume - SMS-EMOA.	90
5	Algoritmo de redução de indivíduos apenas com hipervolume.	91
6	Algoritmo de redução de indivíduos com dois critérios de seleção.	92
7	Algoritmo DE original.	95
8	Algoritmo evolucionário baseado em decomposição - MOEAD.	98
9	Algoritmo Transgenético Geral.	103
10	Algoritmo de melhoria por Substituição de Arestas.	111
11	Algoritmo de busca local baseado em ciclos.	113
12	Algoritmo de Construção de Árvore para o ACO.	115
13	Algoritmo de Escolha de Próximo Componente - proximo_passo	116
14	Algoritmo Baseado em Colônia de Formigas.	116
15	Algoritmo para criação de árvores baseado em árvore geradora do GRASP adaptado de (MARTINS et al., 1999).	118
16	Algoritmo de Construção de Árvore baseado em Caminho mais Curto do GRASP.	119
17	Algoritmo GRASP proposto.	119
18	Algoritmo de Construção de Solução para Algoritmo Genético.	121
19	Algoritmo de Recombinação de solução	121
20	Algoritmo de mutação do algoritmo genético.	122
21	Algoritmo Genético proposto.	123
22	Algoritmo para construção/perturbação de solução do ILS.	124
23	Algoritmo de Busca Local do ILS	125
24	Algorithm ILS proposto.	126
25	Heurística Multi-Stream Multi-Source Voronoi Diagram - MMVD	129
26	Algoritmo de criação de solução.	131
27	Algoritmo de recombinação com base no custo de árvore.	132
28	Algoritmo de recombinação por caminho.	133
29	Algoritmo de recombinação por hops.	134
30	Algoritmo de recombinação diferencial.	135
31	Algoritmo PathPlasmídio	137
32	Algoritmo TreePlasmídio	138
33	Algoritmo Transgenético Proposto - TA.	141

Sumário

1	INTRODUÇÃO	37
1.1	Objeto de Estudo	37
1.2	Motivação	39
1.3	Metodologia	40
1.4	Objetivos	41
1.5	Contribuições	42
1.6	Organização do trabalho	44
2	TECNOLOGIA <i>MULTICAST</i>	45
2.1	Definições	45
2.1.1	<i>Multicast</i>	45
2.1.2	Grupo <i>Multicast</i>	47
2.1.3	Endereçamento <i>Multicast</i>	47
2.1.3.1	Distribuição de Endereços	47
2.1.3.2	Mapeamento IP <i>Multicast</i> -Endereço MAC	48
2.2	Árvores de distribuição	48
2.2.1	Árvores Baseadas em Fonte	49
2.2.2	Árvores Compartilhadas	49
2.2.3	Construção de Árvores de Distribuição	50
2.3	Protocolos de Roteamento <i>Multicast</i>	51
2.3.1	Protocolos intra-domínio	52
2.3.1.1	<i>Internet Group Management Protocolo</i> - IGMP	52
2.3.1.2	<i>Distance Vector Multicast Routing Protocol</i> - DVMRP	52
2.3.1.3	<i>Protocol Multicast Independent</i> - PIM	52
2.3.2	Protocolos inter-domínio	53
2.3.2.1	<i>Multicast Source Discovery Protocol</i> - MSDP	53
2.3.2.2	<i>Multiprotocol Border Gateway Protocol</i> - MBGP	53
2.3.2.3	<i>Source Specific Multicast</i> - SSM	54
2.3.3	Classificação dos Protocolos de Roteamento	55
2.4	Qualidade de Serviço	55
2.5	Resumo	56
3	ROTEAMENTO <i>MULTICAST</i> EM GRAFOS	59
3.1	Roteamento <i>Multicast</i>	59
3.1.1	Definição do problema	59
3.1.2	Representação em Grafos	59

3.1.3	Generalizações do Problema de Roteamento <i>Multicast</i>	60
3.2	Problema de Roteamento <i>Multicast</i> Mono-Sessão	61
3.2.1	Abordagens Mono-Fonte	61
3.2.2	Abordagem Multi-Fonte	62
3.3	Problema de Roteamento <i>Multicast</i> Mono-Sessão Multiobjetivo	63
3.3.1	Abordagens	63
3.4	Roteamento <i>Multicast</i> Multissessão	65
3.4.1	Congestão	65
3.4.2	Capacidade Residual	68
3.4.2.1	Capacidade Residual Sujeito a Limite de Crescimento das Árvores	68
3.4.2.2	Capacidade Residual Sujeito com Suporte para Múltiplas Fontes	70
3.4.3	Custo de Configuração	72
3.4.3.1	Otimização do Custo com Limite de Capacidade das Arestas	72
3.4.3.2	Otimização do Custo com Limite de <i>Delay</i>	73
3.4.3.3	Otimização de Custo com Limite <i>Hops</i>	74
3.5	Outras Abordagens	75
3.6	Resumo	76
4	OTIMIZAÇÃO MULTIOBJETIVO	77
4.1	Noções Básicas	77
4.1.1	Problema Multiobjetivo	77
4.1.2	Relação de Dominância	78
4.1.3	Conjuntos de Aproximação	79
4.1.4	Conjuntos de Aproximação e Relação de dominância	80
4.2	Avaliação de Conjuntos de Aproximação	80
4.2.1	<i>Rank</i> de Dominância	80
4.2.2	Indicador de Qualidade	81
4.2.2.1	Hipervolume	82
4.2.2.2	Epsilon	83
4.3	Algoritmos Evolucionários	84
4.3.1	Definições	84
4.3.2	Abordagens Clássicas	84
4.3.3	Abordagens Elitistas	87
4.3.3.1	<i>Non-Dominated Sorting Algorithm II</i> - NSGAII	87
4.3.3.2	S-Metric Selection Evolutionary Multiobjective Algorihtm - SMS-EMOA	89
4.3.4	Abordagens não-elitistas	93
4.3.4.1	<i>Generalized Differential Evolution</i> - GDE3	93
4.3.4.2	<i>Multiobjective Evolutionary Algorithm based on Decomposition</i> - MOEA/D	97
4.4	Algoritmos Transgenéticos	99

4.4.1	Fundamentos Biológicos	99
4.4.2	Fundamentos Computacionais	100
4.5	Resumo	103
5	FORMULAÇÃO MATEMÁTICA	105
5.1	Capacidade Residual sob Limite de Orçamento	105
5.2	Roteamento <i>Multicast</i> Multissessão Multiobjetivo	107
5.3	Análise dos Modelos	109
5.4	Resumo	110
6	ALGORITMOS PROPOSTOS	111
6.1	Otimização de Capacidade Residual Sob Limite de Custo	111
6.1.1	Operadores de Busca Local	111
6.1.2	Algoritmo baseado em colônia de formigas	113
6.1.2.1	Construção de Solução	114
6.1.2.2	Escolha de Componente de Solução	115
6.1.2.3	Algoritmo Proposto	116
6.1.3	Algoritmo baseado em GRASP	117
6.1.3.1	Construção de Solução	117
6.1.3.2	Algoritmo Proposto	119
6.1.4	Algoritmo genético	120
6.1.4.1	Representação de Solução	120
6.1.4.2	Operador de Construção de Solução	121
6.1.4.3	Operadores de Recombinação e Mutação	121
6.1.4.4	Algoritmo Proposto	123
6.1.5	Algoritmo baseado em <i>Iterated Local Search</i>	123
6.1.5.1	Construção de Solução	123
6.1.5.2	Busca Local	125
6.1.5.3	Algoritmo Proposto	125
6.2	Heurística MMVD	126
6.2.1	Diagramas de Voronoi	127
6.2.2	Algoritmos Proposto	128
6.2.3	Análise de Complexidade	129
6.3	Algoritmos para Otimização Multiobjetivo	130
6.3.1	Representação de Solução	130
6.3.2	Operador de Criação de Solução	131
6.3.3	Operadores de Recombinação e Mutação	131
6.3.3.1	Recombinação por Árvore	132
6.3.3.2	Recombinação por Custo	132
6.3.3.3	Recombinação por Caminho	132

6.3.3.4	Recombinação por <i>Hops</i>	133
6.3.3.5	Recombinação diferencial	134
6.3.3.6	Consolidação de Árvore	135
6.4	Algoritmo Transgenético	136
6.4.1	Informação Genética	136
6.4.2	Plasmídios e Transposon	137
6.4.3	Subpopulações e Soluções de Elite	139
6.4.4	Implementação	139
6.5	Resumo	143
7	RESULTADOS EXPERIMENTAIS	145
7.1	Instâncias	145
7.1.1	Conjunto de Teste - Mono-Fonte	146
7.1.2	Conjunto de Teste - Multi-Fonte	146
7.1.3	Conjunto de Teste - Multiobjetivo	149
7.2	Plataforma de Execução	149
7.3	Resultados <i>Solver</i> Gurobi	150
7.3.1	Otimização de Capacidade Residual Sujeito a Custo	150
7.4	Resultados do Roteamento Multissessão com Limite de Orçamento	151
7.4.1	Definição dos Valores de Parâmetros	152
7.4.2	Metodologia	153
7.4.3	Avaliações	154
7.5	Resultados do Roteamento Multissessão com Múltiplas Fontes . .	156
7.5.1	Metodologia	156
7.5.2	Cenário 1 - Multissessão Mono-Fonte: $t^k = 1, s = 1$	157
7.5.3	Cenário 2 - Multissessão Multi-Fonte: $t^k = 1, s \geq 5$	158
7.5.4	Cenário 3 - Multissessão Mono Fonte: $t^k \geq 1, s \geq 1$	158
7.5.5	Cenário 4 - Avaliação de Escalabilidade	159
7.6	Resultados do Roteamento Multissessão Multiobjetivo	161
7.6.1	Definição de Valores de Parâmetros	161
7.6.2	Algoritmos	161
7.6.3	Experimento 1 - Avaliação de Instâncias gerais	163
7.6.3.1	Metodologia de Avaliação	163
7.6.3.2	Resultados	166
7.6.4	Experimento 2 - Avaliação de instâncias com Pareto	172
7.6.4.1	Metodologia de Avaliação	172
7.6.4.2	Resultados	173
7.7	Resumo	179
8	CONSIDERAÇÕES FINAIS	181

8.1	Trabalhos Futuros	184
A	RESULTADOS DO <i>modelo_z</i>	187
B	RESULTADOS DO <i>Modelo_{chen}</i>	201
C	RESULTADOS DO MODELO MULTIOBJETIVO EXPERIMENTO 1	211
D	RESULTADOS DO MODELO MULTIOBJETIVO EXPERIMENTO 2	279
	REFERÊNCIAS	321

1 Introdução

1.1 Objeto de Estudo

A demanda por serviços associados às telecomunicações tem crescido de forma espantosa desde o advento das tecnologias que originaram a Internet. Inúmeras aplicações foram desenvolvidas e continuam sendo evoluídas. Novas aplicações aparecem como inovações ou reestruturação de tecnologias já consolidadas. Especificamente relacionado as comunicações via rede, várias aplicações têm lugar de destaque como: IPTV, aplicações relacionadas a conferência usando áudio/vídeo, jogos online, distribuição de conteúdo digital, educação à distância entre outros.

O crescimento do número de aplicações tem aumentado a concorrência por recursos da rede. Em contrapartida, muitas pesquisas têm focado no desenvolvimento de técnicas que atenuem os efeitos da utilização ineficiente dos recursos da rede. Duas tecnologias amplamente difundidas são: *unicast* e *multicast*.

Estas tecnologias atuam na transmissão de informação proveniente de emissor que devem entregar dados a um conjunto de cliente(s). A transmissão via *unicast* tem como base a definição de um caminho entre emissor e receptor. Este caminho inclui roteadores na rede subjacente. Esta é uma das formas mais simples de entregar informação. Entretanto, apresenta alguns problemas no que diz respeito à utilização dos recursos da rede (por exemplo, retransmissão desnecessária de pacotes). Na transmissão via *multicast*, tem-se por objetivo, desde sua concepção, a utilização de forma eficiente da rede evitando, por exemplo, retransmissão de pacotes de forma desnecessária. Para tanto, técnicas de replicação de pacotes são aplicadas de forma controlada evitando cópias de pacotes circulando desnecessariamente pela rede (OLIVEIRA; PARDALOS, 2005).

O *multicast* tem sido amplamente estudado devido suas características relacionadas à construção da estrutura de transmissão. A grosso modo, o processo de transmissão demanda uma estrutura que pode ser vista como uma árvore. Esta árvore é uma agregação dos caminhos entre fonte e clientes(s). Ocasionalmente, os *links* (agregações) de transmissão podem ser compartilhados entre diferentes clientes (neste trabalho *links* serão denominados arestas). Além disso, a tecnologia garante que apenas um pacote será enviado da fonte e o mesmo será copiado apenas quando um determinado nó da árvore possui duas subárvores, cada uma delas levando a cliente(s) diferente(s).

No que diz respeito à construção das árvores de distribuição de informação empregadas no *multicast*, duas técnicas são amplamente utilizadas: **árvores baseadas em fonte** (*source-based trees*), que possui uma árvore enraizada em um emissor de informação; **árvores compartilhadas** (*shared-based trees*), que devem definir um ponto na rede que

esteja próximo aos clientes e emissor por alguma métrica especificada. Este ponto é denominado *Rendezvous Point* e é responsável por receber informações de/para o emissor e retransmitir de/para os clientes. Em estudo realizado por Chen, Gunluk e Yener (2000), os autores indicam que árvores baseadas em fonte utilizam menos arestas (*links*) que a outra forma mencionada.

O processo de roteamento via *multicast* tem como núcleo a criação de árvores de distribuição. Inúmeros algoritmos são aplicados para tal tarefa. Levando-se em conta as duas abordagens mencionadas no parágrafo anterior, duas técnicas são amplamente difundidas: construção de árvores de distribuição por meio de árvores de Steiner (HWANG; RICHARDS; WINTER, 1992); construção de árvores baseadas em núcleo. Neste caso, dois problemas são encontrados: primeiro deve-se definir qual seria o melhor componente da rede e, em seguida, construir a árvore de distribuição.

A construção de árvores de distribuição *multicast* pode ser modelada como um problema de otimização em grafos (KOMPELLA; PASQUALE; POLYZOS, 1993) e (OLIVEIRA; PARDALOS, 2005). Este problema tem sido amplamente estudado com diferentes enfoques na modelagem e projeto de algoritmos.

O modelo mais simples e comum trata o problema com apenas uma árvore de distribuição cujo objetivo é otimizar o custo de configuração da mesma (OLIVEIRA; PARDALOS, 2005). Resende e Pardalos (2006) apresentam o problema sob uma visão mono-objetivo visando a otimização do custo de configuração da solução. Inúmeras técnicas foram e tem sido apresentadas para o problema neste modelo. Entretanto, conforme as soluções foram melhoradas, novas demandas foram incorporadas ao problema. Elas estão associadas a requerimentos de aplicações específicas. Em geral, tais demandas estão relacionadas a qualidade de serviço (QoS).

Os trabalhos pioneiros consideram o problema levando em conta a construção de apenas uma árvore de distribuição. Entretanto, há cenários nos quais existe a demanda por múltiplas árvores de distribuição *multicast*. Chen, Gunluk e Yener (2000) apresetaram uma nova abordagem para lidar com o problema com múltiplas sessões *multicast* e, consequentemente, múltiplas árvores de distribuição. Os autores utilizaram a congestão da rede como métrica e a função objetivo visava a redução da congestão. Vários trabalhos subsequentes melhoraram os resultados do trabalho de (CHEN; GUNLUK; YENER, 2000). Além disso, introduziram novos elementos aos modelos matemáticos como é caso de (LEE; CHO, 2004) e (WANG; LIANG; JAN, 2002).

A utilização de múltiplas árvores tem sido empregada para entrega de conteúdo em rede P2P. Em (ZHENG; CHO; XIA, 2015), propõe-se um modelo que visa a otimização da taxa de transferência alcançada por cada árvore. O objetivo é maximizar este valor. Além disso, há também trabalhos relacionados ao consumo de energia existente nos dispositivos que integram o sistema de comunicação via *multicast*. Em (WANG et al., 2015), tem-se como objetivo a otimização visando ao baixo consumo de energia, ao que se denomina

green networks.

O problema de roteamento *multicast* tanto na versão com uma sessão quanto na versão multissessão é NP-Completo. Em (OLIVEIRA; PARDALOS, 2005), apresenta-se uma prova para o caso com apenas uma sessão. Em (CHEN et al., 2013), apresenta-se uma prova para o caso de otimização da capacidade residual com múltiplas sessões.

Como pode ser observado, na literatura há lacunas importantes com relação as abordagens empregadas na resolução do problema na versão com mais de uma sessão - multissessão. Apenas um trabalho faz uso das abordagens meta-heurísticas como é o caso do trabalho de (RANDACCIO; ATZORI, 2007). Há espaço para estudos considerando abordagens meta-heurísticas importantes e que obtiveram sucesso em outros problemas.

Além disso, os trabalhos mais importantes podem ser classificados em duas categorias com respeito ao objetivo de otimização: custo e capacidade residual (medida de congestionamento da rede). Entretanto, nenhum trabalho realizou uma investigação sobre um modelo que considere a otimização de recursos da rede como capacidade residual aliada a otimização do custo (desconsiderando a aplicação de penalidade do modelo de Chen, Gunluk e Yener (2000)).

Diante do exposto, percebe-se que há possibilidade de investigação de um modelo multiobjetivo para o problema de roteamento *multicast* multissessão. Assim sendo, *o objeto de estudo desta proposta é o problema de roteamento multissessão multiobjetivo. Especificamente, os objetivos de otimização são capacidade residual, custo e, adicionalmente, contagem de hops. Além disso, também são realizados estudos de modelos relacionados a proposta multiobjetivo. Um modelo otimizando capacidade residual sujeito a custo e um modelo com múltiplas fontes por sessão.*

1.2 Motivação

O homem tem se tornando cada vez mais dominante na natureza. Sua principal ferramenta tem sido a capacidade de planejar, executar e avaliar. Assim sendo, todo processo produtivo que venha a ter sucesso, inevitavelmente, precisa de um bom planejamento. O planejamento visa a construção de soluções viáveis para serem aplicadas a situações reais. Entretanto, inúmeros elementos do contexto real podem ser ignorados, permitindo que se foque apenas no que é de real importância para resolução do problema em questão.

Uma representação simples da realidade abordada permite que se modele, artificialmente, situações reais nos mais diversos segmentos de atuação humana. Isso permite que os processos sejam otimizados e os recursos sejam alocados de forma mais eficiente possível. Especificamente, a pesquisa operacional tem sido uma ferramenta importante na otimização de recursos nos mais diferentes segmentos sociais tais como: extração de petróleo, logística, otimização de uso de matéria-prima entre outros. Para tanto, uma modelagem significativa faz-se necessário. Logo, quanto mais representativo o modelo for

em relação a realidade, mais eficientes podem ser as soluções encontradas. Vale ressaltar que a modelagem deve evitar que o modelo proposto se torne muito difícil a ponto de inviabilizar uma solução viável (GOLDBARG; LUNA, 2005).

Visando a proximidade com o contexto de aplicação real, muitos pesquisadores têm adotado modelos matemáticos multiobjetivo para os problemas estudados. Esta abordagem propicia a otimização em um contexto que possui demandas conflitantes e, possivelmente, com mesma importância. Especificamente, o objeto de estudo deste trabalho apresenta situação análoga ao considerar a otimização de funções-objetivo conflitantes como é caso da capacidade residual, custo e contagem de *hops*.

O problema de roteamento *multicast* possui aplicação em diversos tipos de serviços de entrega de dados. Este fator tem motivado diferentes pesquisas sobre como prover rotas eficientes de acordo com os requisitos da aplicação. A lacuna verificada na pesquisa em respeito à abordagem motivou a presente pesquisa. Como presente trabalho vai comprovar, o problema abordado na versão multissessão possui características propícias a abordagem multiobjetivo, o que de fato não se considerou ainda. Muitos problemas de otimização combinatória têm sido modelados e resolvidos sob uma ótica multiobjetivo com excelentes resultados (STEWART et al., 2008)

Além disso, o problema de roteamento *multicast* na versão objeto de estudo deste trabalho permite a aplicação de algoritmos clássicos para o problema da árvore de Steiner considerando otimização de custo, assim como otimização de custo sujeito a limite de *hops* (VOSS, 1999). Estas heurísticas podem ser facilmente adaptadas para construção de soluções para aplicação ao problema na versão multiobjetivo.

Por fim, um fator importante para aplicação da abordagem multiobjetivo é a possibilidade de criar um modelo mais geral e que permita otimizar demandas conflitantes. Isso pode ser feito dando a mesma importância para todos os objetivos. Especificamente, no modelo proposto na Seção 5.2 é possível contemplar três abordagens importantes presentes na literatura como é o caso de (LEE; CHO, 2004; CHEN et al., 2013) que tratam da otimização da capacidade residual com sessões multi-fonte. Além disso, a versão multiobjetivo incorpora o modelo proposto por Kang, Park e Park (2009).

Resumidamente, os fatores motivantes para este trabalho foram:

- O problema ser relevante no cenário de otimização combinatória. Isso se deve ao fato do problema ter várias e significativas aplicações reais;
- A revisão da literatura comprova que a versão multiobjetivo do modelo, considerando o custo e a capacidade residual não foi abordada. Alguns trabalhos modelam o problema adicionando restrições que podem ser incorporadas como função-objetivo;
- É possível adaptar diferentes arquiteturas gerais de otimização multiobjetivo para resolução do problema aliado à adaptação de heurísticas.

1.3 Metodologia

Este trabalho foi dividido em três etapas. A primeira parte consiste em uma pesquisa predominantemente bibliográfica (GIL, 2010). As fontes utilizadas para realizar a pesquisa bibliográfica são trabalhos aceitos em periódicos internacionais e teses. Especificamente, serão avaliados os trabalhos presentes em periódicos científicos, teses e dissertações, anais de congresso, periódicos de indexação e resumo (GIL, 2010). Para facilitar a consulta, bases de dados serão utilizadas, entre elas destacam-se: IEEE, ScienceDirect, Springer e ACM. Além disso, os sistemas de buscas convencionais da Web serão utilizados. Para seleção dos artigos, foi realizada uma leitura exploratória, através de resumos e palavras chaves indexadas. Em seguida, os artigos mais relevantes serão lidos de forma seletiva, pois estes devem estar em conformidade com os objetivos da pesquisa. Por fim, uma leitura mais profunda será realizada.

Na segunda etapa, será realizada a implementação das principais abordagens de representação de solução, considerando as abordagens selecionadas na revisão bibliográfica. A segunda etapa consiste na implementação dos algoritmos para os modelos projetados e também para os modelos estudados na literatura. Esta etapa consiste na definição das representações de soluções, operadores e algoritmos baseados em abordagens da literatura.

Por fim, uma análise experimental é realizada para avaliar os algoritmos diante dos modelos propostos. Esta etapa consiste na definição das instâncias empregadas nos experimentos. Em seguida, os parâmetros de cada algoritmo foram obtidos por meio do IRACE (CÁCERES; LÓPEZ-IBÁÑEZ; STÜTZLE, 2014). Em seguida, todos os experimentos necessários as avaliações dos algoritmos foram realizados. Estes experimentos foram conduzidos de acordo com metodologias definidas conforme explicado no Capítulo 7. O resultados obtidos foram analisados por meio de testes não-paramétricos e as conclusões finais foram derivadas do método apresentado no Capítulo 7.

1.4 Objetivos

Os objetivos deste trabalho são:

- Analisar os modelos de solução da literatura e propor novos modelos para o problema de roteamento *multicast* especialmente nos seguintes temas:
 - modelo mono-objetivo visando a otimização de capacidade residual sujeito a custo;
 - modelo com múltiplas fontes por sessão otimizando capacidade residual;
 - modelo multiobjetivo que incorpora demandas de qualidade de serviço e otimiza custo, capacidade residual e contagem de *hops*.

- projetar algoritmos para o modelo mono-objetivo utilizando meta-heurísticas diversificadas;
- projetar algoritmo heurístico para tratar o problema com múltiplas fontes;
- projetar algoritmo transgenético para aplicação ao problema na versão multiobjetivo;
- adaptar *frameworks* multiobjetivo para aplicação ao problema na versão multiobjetivo;
- realizar avaliações dos algoritmos propostos utilizando metodologia consistente por meio de testes estatísticos e conjuntos de instâncias com características diferentes;

1.5 Contribuições

No presente trabalho, foi realizado amplo estudo sobre o problema de roteamento *multicast*. O foco principal foram os modelos concebidos ao longo dos anos e os algoritmos projetados. Como resultado, apresenta-se uma revisão sobre o problema de roteamento *multicast* com relação ao uso de múltiplas sessões na mesma aplicação. O resultado deste trabalho foram as avaliações sobre os diferentes tipos de abordagens já estudadas, assim como a descoberta de lacunas importantes.

O levantamento bibliográfico mostrou que algumas lacunas com relação ao problema de roteamento *multicast* multi-sessão ainda existiam. Por exemplo, praticamente não há estudos que apliquem abordagens baseadas em meta-heurísticas. Além disso, nenhum trabalho considerou a utilização de modelos multiobjetivo.

Neste trabalho, dois modelos matemáticos são propostos visando a otimização de objetivos existentes na literatura do problema. Em geral, estes objetivos são tratados de forma isolada. Os trabalhos em geral levam em consideração apenas um deles. Por exemplo, a otimização da capacidade residual é um objetivo utilizado em vários trabalhos. Em contrapartida, não se considera o custo. Do mesmo modo, trabalhos que definem o custo como prioridade não incorporam a capacidade residual como parte do modelo. O primeiro modelo proposto visa otimizar a capacidade residual impondo um limite de custo à construção da solução. Este modelo é uma iniciativa para tratar objetivos conflitantes simultaneamente. O segundo modelo é uma versão multiobjetivo do problema de roteamento *multicast* multissessão. Neste caso, três objetivos foram considerados: custo, capacidade residual e contagem de *hops*.

O modelo multiobjetivo é inovador com relação aos modelos já existentes. As soluções de Pareto para uma dada instância incorporam soluções de modelos importantes da literatura como é o caso dos modelos proposto por (LEE; CHO, 2004), (WANG; LIANG; JAN, 2002) e (KANG; PARK; PARK, 2009). O modelo proposto por Lee e Cho (2004)

considera a otimização da capacidade residual restrita ao crescimento das árvores em termos de número de arestas. O modelo proposto por Wang, Liang e Jan (2002) considera a otimização do custo de configuração usando um limite de capacidade das arestas. O trabalho de Kang, Park e Park (2009) considera a otimização do custo de configuração sujeito a contagem de *hops* entre a fonte e os clientes para cada árvore de distribuição. O modelo de Chen et al. (2013) também pode ser considerado se for excluído o tratamento de múltiplas fontes por sessão. As soluções deste modelo podem ser alcançadas considerando a otimização de Pareto e as funções-objetivo do modelo proposto. Além disso, os objetivos de otimização considerados contemplam boa parte dos trabalhos da literatura e incluem os aspectos relacionados a qualidade de serviço, que é demanda de muitas aplicações *multicast*.

Com relação ao projeto de algoritmos, várias propostas foram implementadas para os três modelos estudados neste trabalho (dois modelos mencionados no parágrafo anterior e um terceiro modelo da literatura). Os algoritmos foram baseados em arquiteturas já consolidadas como é o caso das meta-heurísticas baseados em colônia de formigas, o GRASP (*Greedy Randomized Adaptive Search Procedure*), algoritmo genético e ILS (*Iterated Local Search*). Estes algoritmos foram aplicados ao problema definido no modelo que otimiza capacidade residual sujeito a limite de custo. Além disso, também foi projetada uma heurística para o modelo proposto por Chen et al. (2013). Esta heurística foi baseada nos diagramas de Voronoi e obteve excelentes resultados contra procedimentos da literatura em diversas situações experimentais. Por fim, foram projetados algoritmos para o modelo multiobjetivo. Cinco algoritmos adaptados a partir de arquiteturas de otimização multiobjetivo foram construídos. Além disso, foi proposto um algoritmo transgenético para o problema na versão multiobjetivo.

O algoritmo transgenético proposto utiliza o método de classificação de fronteiras de Pareto originalmente implementado no trabalho de Deb et al. (2002). Além disso, há uma nova proposta de construção e manutenção de subpopulações. Estas subpopulações evoluem separadamente obtendo informações genéticas de bases de dados compartilhadas. Cada subpopulação é caracterizada por um tipo de construtor de solução, que é direcionado para um dos objetivos do problema. Diferentes versões do algoritmo foram avaliadas.

Por fim, um estudo algorítmico foi realizado para avaliar os algoritmos propostos para cada um dos modelos estudados neste trabalho. Todas as avaliações possuem um metodologia bem definida. A metodologia consiste na definição de instâncias, afinamento de parâmetros dos algoritmos, coleta das amostras e aplicação de testes estatísticos. Em todas as avaliações foi considerado o teste estatístico apropriado e uso de avaliação *posthoc* para averiguar onde havia diferença entre pares de algoritmo. Este último teste foi utilizado para que fosse realizada contagens das avaliações nas quais um algoritmo teve resultado estatisticamente superior a outro. Assim sendo, foi possível observar o aproveitamento

percentual de cada algoritmo nas avaliações. Nas avaliações dos modelos mono-objetivo considerou-se o valor da função objetivo, custo de configuração e tempo de processamento. Para o caso do modelo multiobjetivo, foi considerado o uso dos indicadores de qualidade $I_{\epsilon+}$, I_{ϵ} e I_H , assim como o tempo de processamento.

Como produção científica associada à presente pesquisa destacam-se as seguintes publicações:

- ANDRADE, R. C.; GOLDBARG, M. C. ; GOLDBARG, E. F. G. . *Bi-Objective Multicast Packing Problem*. Anais do XLV Simpósio Brasileiro de Pesquisa Operacional, v. 1, p. 1904-1915, 2013.
- ANDRADE, R. C.; GOLDBARG, M. C. ; GOLDBARG, E. F. G. . *Roteamento Multicast com Múltiplas Sessões Sujeito a Limite de Orçamento*. In: Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional, 2015. Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional, 2015. p. 2258-2269.
- ANDRADE, R. C.; GOLDBARG, M. C.; GOLDBARG, E.F.G. *An Algorithm Based on Voronoi Diagrams for the Multi-Stream Multi-Source Multicast Routing Problem*. International Journal of Innovative Computing and Applications. (**Aprovado para publicação**).

1.6 Organização do trabalho

O presente trabalho está dividido nos seguintes capítulos. Capítulo 1, que introduz o trabalho, objeto de estudo e resultados já alcançados. No Capítulo 2, apresenta-se uma revisão sobre a tecnologia *multicast* com foco nas técnicas de construção de árvores de distribuição. No Capítulo 3, é apresentada uma revisão sobre o problema de roteamento *multicast* em grafos. Na revisão realizada, considera-se o problema sob vários aspectos incluindo as versões mono-objetivo e multiobjetivo tanto para os casos com apenas uma sessão quanto para os casos com multissessões. No Capítulo 4, uma revisão sobre otimização multiobjetivo é apresentada considerando os fundamentos básicos da teoria desde comparação de soluções à comparação de conjuntos de aproximação por meio de indicadores de qualidade. No Capítulo 5, os modelos matemáticos propostos são descritos em detalhes e é feita uma análise sobre os mesmos. No Capítulo 6, todos os algoritmos projetados e aplicados aos modelos propostos e ao modelo proposto por (CHEN et al., 2013) são apresentados. Os algoritmos são divididos por seção, onde cada uma delas foca em um dos modelos considerados. No Capítulo 7, todo o processo de experimentação é apresentado. O processo considera a criação dos casos de teste, definição de parâmetros de algoritmos, coleta de amostras e análises estatísticas. Por fim, no Capítulo 8, são apresentadas as conclusões finais deste trabalho.

2 Tecnologia *Multicast*

Os principais conceitos referentes a tecnologia *multicast* são descritos neste capítulo. Apresenta-se inicialmente, os fundamentos da tecnologia, na Seção 2.1. Em seguida, o mecanismo de distribuição de conteúdo via árvores de distribuição, na Seção 2.2. No restante do capítulo, os protocolos de comunicação *multicast* são apresentados e sua relação com as árvores de distribuição, na Seção 2.3. O capítulo é finalizado considerando os aspectos relacionados à qualidade de serviço na transmissão de dados em aplicações multimídia, na Seção 2.4.

2.1 Definições

Nesta seção, são considerados os elementos mais essenciais: conceitos básicos, endereçamento e gerenciamento de grupos. Eles estão presentes em todas as camadas do modelo OSI (ZIMMERMANN, 1980). Outro elemento essencial é o roteamento discutido em detalhes no capítulo 3 como um problema de roteamento em grafos.

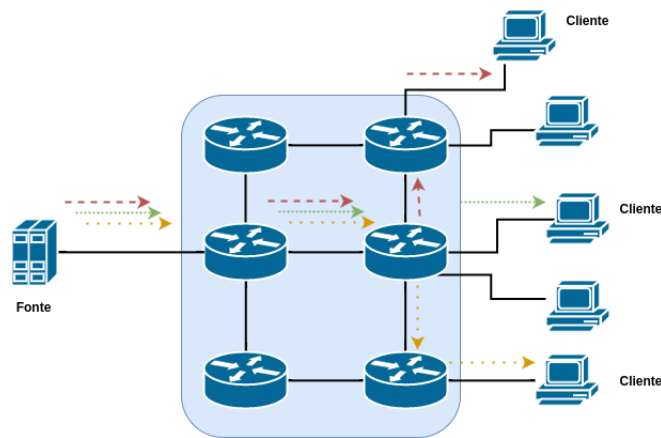
2.1.1 *Multicast*

O *multicast* é uma tecnologia empregada na comunicação em redes de computadores. Esta tecnologia tem como princípio básico a economia de banda da rede. A comunicação dá-se através da utilização de grupos de *hosts*. A ideia central do funcionamento do *multicast* é fazer a transmissão de *streams* de dados para *hosts* interessados. O *stream* é enviado por um *host* do sistema que assume o papel de emissor de informação e é denominado **fonte do serviço**. Os *hosts* receptores do stream (interessados no serviço) são identificados como **clientes do serviço** (também chamados de vértice de destino).

A transmissão de informação via *multicast* é realizada mediante rotas estabelecidas entre a fonte do serviço e os clientes. Estas rotas são estabelecidas de modo a otimizar o uso da rede. Um dos recursos básicos do *multicast* é a capacidade de evitar duplicação desnecessária de pacotes ao longo das rotas – como ocorre em outras abordagens (*unicast* e *broadcast*).

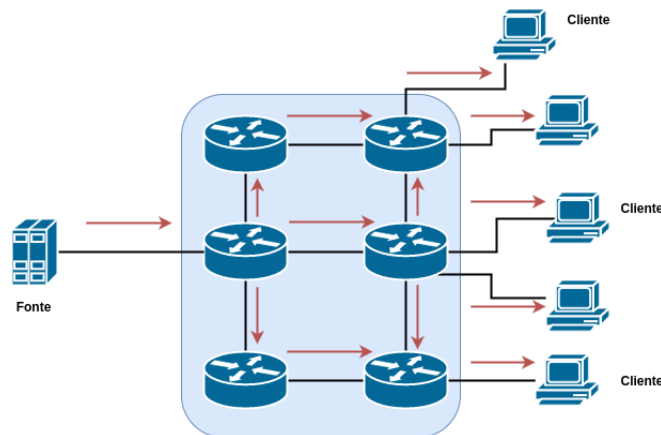
As abordagens *unicast* e *broadcast* em comparação ao *multicast* apresentam alto consumo de banda na rede visto que não foram projetadas com o intuito de preservar tal recurso. A seguir, nas Figuras 1, 2 e 3 ilustra-se tal situação.

Na Figura 1, um exemplo de utilização do *unicast* é apresentado. Observe-se que o mesmo pacote de dados é enviado para três clientes diferentes. Não há utilização de um mecanismo que evite duplicação de pacotes, desse modo subutilizando os recursos da rede.

Figura 1 – Exemplo de fluxo de dados em uma rede IP.

Fonte: própria.

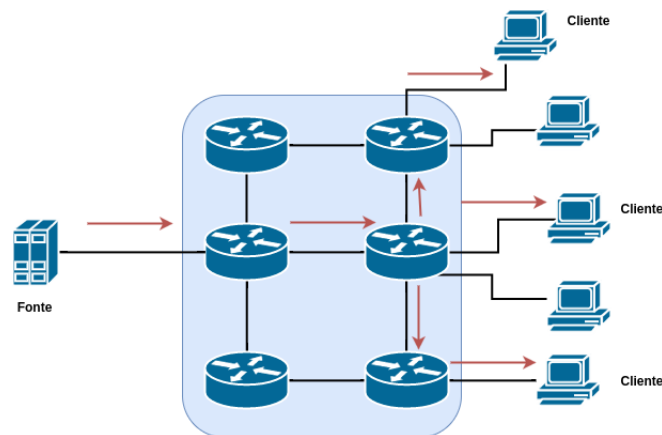
O mesmo ocorre na utilização de *broadcast*, conforme ilustrado na Figura 2. Neste exemplo, o mesmo pacote de dados é duplicado em todas as saídas de roteadores e atinge *hosts* não interessados na informação. A situação agrava-se diante da transmissão de vários pacotes de dados diferentes.

Figura 2 – Exemplo de fluxo de dados *broadcast*.

Fonte: própria.

Em oposição ao cenário anterior, o *multicast* apresenta mecanismos para evitar subutilização da rede de forma transparente conforme ilustrado na Figura 3. Os pacotes são enviados apenas para os clientes do serviço, o que significa que o pacote de dados não é duplicado desnecessariamente em qualquer roteador ao longo das rotas.

Por fim, vale ressaltar que é necessário que os roteadores responsáveis pelo roteamento da informação tenham capacidade para *multicast*.

Figura 3 – Exemplo de fluxo de dados em uma rede com IP *multicast*.

Fonte: própria.

2.1.2 Grupo *Multicast*

Um grupo *multicast* é constituído de um conjunto de clientes de um serviço (*hosts*). Essa informação é enviada por uma fonte do serviço através de distribuição *multicast*. Além disso, um grupo possui uma identificação realizada através de um endereço IP *Multicast*. Os endereços utilizados para definir um grupo *Multicast* são pertencentes a classe D no IPv4. Para receber a informação via *Multicast* é necessário juntar-se ao grupo *Multicast* em questão. Isto é realizado por meio de um protocolo específico: *Internet Group Managing Protocol* (IGMP) no IPv4 ou *Multicast Listener Delivery* (MLD) no IPv6.

2.1.3 Endereçamento *Multicast*

2.1.3.1 Distribuição de Endereços

O endereçamento *Multicast* é parte fundamental da tecnologia. A comunicação entre fonte e clientes é permitida por meio dos endereços **IP *Multicast***. Além disso, deve-se fazer o mapeamento entre o *Multicast* IP e o endereço MAC dos clientes da informação fornecida.

A *Internet Assigned Numbers Authority* (IANA) definiu o espaço de endereço da classe D para comunicação *Multicast*. Especificamente, a faixa de endereços entre 224.0.0.0 e 239.255.255.255, considerando o *Multicast* IPv4. Os endereços na faixa 224.0.0.0 a 224.0.0.255 são aplicados a redes locais (IETF, 2017a). A faixa de endereços 224.0.1.0 a 238.255.255.255 incluem dois tipos de endereços importantes: endereços GLOP e endereços para *Source-Specific Multicast* (SSM). Os endereços GLOP estão associadas aos Sistemas Autônomos (AS - *Autonomous System*), um AS possui identificador que permite

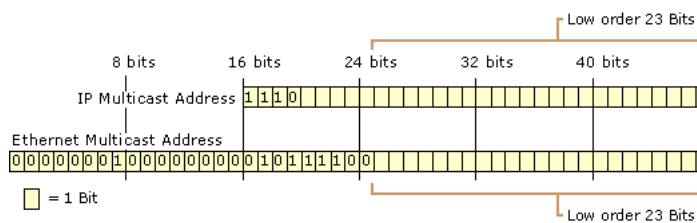
que o mesmo se comunique com outros AS's e possui uma política única de roteamento. O protocolo SSM será discutido na seção 2.3.2.3 com mais detalhes.

2.1.3.2 Mapeamento IP *Multicast*-Endereço MAC

A camada de enlace do modelo OSI trata o envio/recebimento de pacotes via *unicast* e *multicast* de forma diferente. Um pacote enviado via *unicast* possui seu endereço de destino como sendo um endereço MAC do receptor do pacote. Por outro lado, quando um pacote IP *Multicast* é enviado, ele é destinado a um grupo de receptores.

A IANA definiu um mapeamento que permite associar um endereço IP *Multicast* a um endereço MAC. Na Figura 4, é ilustrado um exemplo do mapeamento.

Figura 4 – Associação entre endereço IP *multicast* e endereço MAC.



Fonte: própria.

Como pode ser observado na Figura 4, os 23 bits de mais baixa ordem são diretamente mapeados para o endereço MAC. Os 4 primeiros bits do endereço IP *Multicast* são reservados segundo convenções da classe de endereçamento D. Cinco bits não são mapeados: do 23º bit até o 28º. Assim sendo, no exemplo ilustrado, o 24º do endereço MAC ficou definido como 0. Os 4 bits de mais alta ordem do endereço IP *Multicast* foram copiados a partir do 25º bit do endereço MAC.

2.2 Árvores de distribuição

Uma árvore de distribuição *multicast* consiste na estrutura formada pelas rotas entre fonte e clientes do serviço na rede subjacente. Através da árvore de distribuição, os dados são encaminhados. Essencialmente, uma árvore de distribuição *multicast* é formada por fonte (emissor do stream de dados) e clientes do serviço (receptores). Há dois tipos bastante comuns: Árvores baseadas em fonte (*source-based trees*) e Árvores compartilhadas (*shared-based trees*).

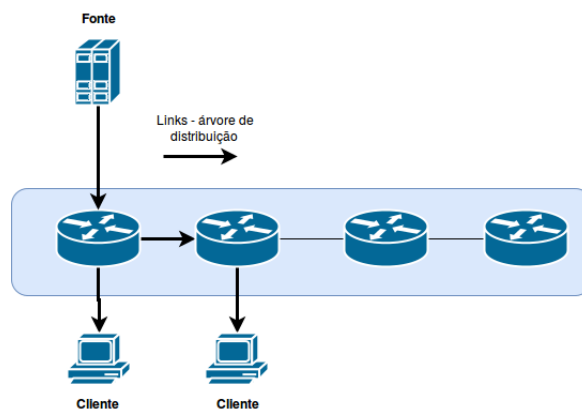
2.2.1 Árvores Baseadas em Fonte

Este tipo de estrutura de distribuição de conteúdo, como o próprio nome sugere, tem uma árvore enraizada na fonte de transmissão. Em geral, ela é formada a partir de caminhos mais curtos - considerando algum parâmetro que permita aferir tais caminhos como custo de comunicação. Devido a isso, esse mecanismo de distribuição de dados é por vezes denominado de árvore de caminho mais curto. Por ser enraizada na fonte, o mecanismo de distribuição por árvore baseada em fonte leva a necessidade de configurar uma árvore por fonte.

A notação adotada para representar a associação entre uma fonte e um grupo de *hosts* é denotada por (S,G) , onde S representa a fonte e G representa o grupo *multicast*, respectivamente endereços IP. Essa informação é armazenada nos roteadores. Isso leva ao aumento na demanda por memória diante do crescimento do número de fontes. Entretanto, um fator positivo nesse tipo de estrutura é a possibilidade de encontrar árvores próximas do ótimo (considerando custo, por exemplo).

Na Figura 5, há uma representação da distribuição de pacotes via árvore baseada em fonte. As arestas mais densas e com setas ilustram a rota pela qual os pacotes chegam aos destinatários.

Figura 5 – Árvore de distribuição *Multicast*.



Fonte: própria.

2.2.2 Árvores Compartilhadas

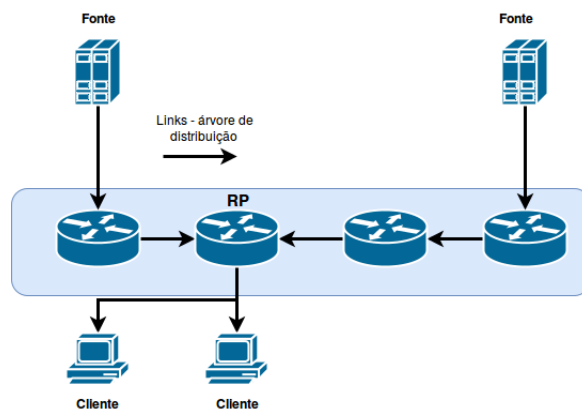
Este mecanismo de distribuição também faz utilização de fontes de transmissão. Entretanto, há um outro elemento importante para encaminhamento dos pacotes de dados. Trate-se do *Rendezvous Point* (RP), que consiste de um nó da rede escolhido de forma estratégica. Em geral, deve possuir um posicionamento que permite melhorar o desem-

penho da aplicação. O RP recebe os pacotes de dados da(s) fonte(s) e os envia para os clientes do serviço.

Um aspecto importante neste tipo de estrutura de comunicação é a localização do RP, tendo em vista que sua utilização se justifica no fato de facilitar a comunicação otimizando a transmissão. O RP pode receber pacotes de dados de diferentes fontes, sendo útil em aplicações onde todos os membros do grupo transmitem dados. A notação empregada é $(*, G)$, onde o $*$ indica que a informação pode vir de fontes diferentes e G representa os destinatários.

Na Figura 6, há um exemplo de transmissão de dados utilizando árvores compartilhadas. As ligações mais densas e com setas ilustram as rotas. Um detalhe importante é o fato de as fontes enviarem os pacotes para o RP, que os encaminha para os destinatários.

Figura 6 – Transmissão em Árvore Compartilhada.

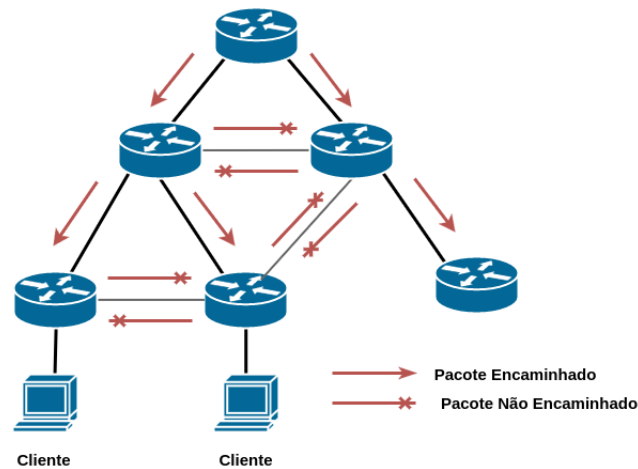


Fonte: própria.

2.2.3 Construção de Árvores de Distribuição

As árvores de distribuição *multicast* são estruturas obtidas a partir da rede subjacente. Algoritmos são empregados para criação de rotas entre a fonte do serviço e os *hosts* interessados. Os protocolos de roteamento *multicast* empregam técnicas de construção de árvores. Entre as principais abordagens estão: *Reverse Path Forwarding* (RPF), *Truncated RPF* (TRPF) e o *Reverse Path Multicasting* (RPM).

O RPF foi uma das primeiras abordagens utilizadas para criação de árvores de distribuição. Seu funcionamento é bastante eficiente e sua implementação é simples. Além disso, este algoritmo previne a criação de ciclos e o encaminhamento desnecessário de pacotes pela rede. O funcionamento do algoritmo ocorre da seguinte forma: um roteador recebe um pacote com o endereço da fonte de transmissão e o encaminha por suas interfaces de saída. Se o mesmo pacote chegar novamente por outra interface ("mais distante"), então o roteador não o encaminha conforme ilustrado na Figura 7.

Figura 7 – Funcionamento do Algoritmo RPF.**Fonte:** própria.

Na Figura 7, a criação de uma árvore de distribuição utilizando o RPF é ilustrada. Observa-se que os pacotes não são encaminhados quando o roteador já encaminhou uma cópia em um momento anterior. Quando isso acontecer, a aresta não será utilizada. Em uma representação por meio de grafos, tal aresta não seria considerada. As linhas mais densas na figura representam a árvore obtida.

O RPF não trata de roteadores que não estão conectados a algum *host* destinatário. Na Figura 7, há uma situação na qual o roteador recebe pacote e está incluso na árvore de distribuição, embora ele seja o ponto final da rota. Essa situação pode se agravar considerando uma árvore de distribuição contendo milhares de roteadores.

Neste cenário, entra em cena o algoritmo TRPF. Ele é uma extensão do algoritmo RPF e tem por objetivo evitar rotas que não são utilizadas, conforme mencionando sobre o RPF. Na prática, o algoritmo poda a árvore de distribuição obtida, tornando a utilização da rede mais eficiente – isso se dá através da checagem de pacotes nos roteadores.

O algoritmo RPM possui duas etapas e trata da poda da árvore para evitar a inclusão de rotas que não conectam *hosts* clientes. Ele induz os roteadores a trocarem mensagens para verificação de pedidos de inclusão/exclusão de membros de um grupo.

2.3 Protocolos de Roteamento *Multicast*

A comunicação entre *hosts* é auxiliada pelos protocolos de roteamento. Os protocolos controlam vários aspectos da comunicação, incluindo a construção de árvores de distribuição – protocolos de roteamento *multicast*.

Os protocolos de roteamento podem ser classificados em intra-domínio e inter-domínio. Isto se verifica pela forma como a comunicação é realizada. Inúmeras redes no mundo

constituem a Internet – grande rede mundial de computadores. Tais redes são relacionadas a organizações, empresas, universidades e etc.

Os AS's estão presentes na Internet e podem comunicar-se entre si e também podem estabelecer comunicação dentro de seu domínio. Neste sentido, a comunicação entre AS's é denominada inter-domínio e aquela realizada dentro do próprio AS's é denominada intra-domínio.

2.3.1 Protocolos intra-domínio

Os protocolos intra-domínio são aplicados a domínios particulares. Estes protocolos são executados dentro de domínios de um AS.

2.3.1.1 *Internet Group Management Protocolo* - IGMP

O protocolo IGMP é utilizado para controlar a entrada e saída de membros de um grupo *multicast*. Ele é utilizado no IPv4 e em domínios locais (Local Area Network - LAN).

Este protocolo está atualmente na versão 3 definida pela RFC 4604 (IETF, 2017b). A versão 1 do protocolo cria dois tipos de mensagens: *report* e *query*, usadas para indicar que a associação a um grupo foi confirmada e para solicitar entrada em um grupo, respectivamente. A versão 2 do protocolo manteve a compatibilidade com a versão anterior e adicionou a mensagem *leave* para que um cliente comunique imediatamente sua saída de um grupo. Por fim, a versão 3 adicionou o '*source-filtering*' permitindo ao cliente informar ao roteador *multicast* de qual grupo ele deseja receber o tráfego.

2.3.1.2 *Distance Vector Multicast Routing Protocol* - DVMRP

O protocolo DVMRP foi definido na RFC (WAITZMAN, 2017). Ele tem como base o protocolo *Routing Information Protocol* (RIP) (HEDRICK, 2017). O RIP é um protocolo aplicado ao roteamento *unicast* onde a informação é transmitida via *hops* até chegar ao destinatário.

O funcionamento do DVMRP difere do RIP, pois uma árvore de distribuição *multicast* é criada para entrega do tráfego. A construção da árvore de distribuição é realizada por meio de um algoritmo específico. Três algoritmos foram utilizados para construção das árvores de distribuição: RPF e TRPF e RPM. O protocolo na versão 3 utiliza o algoritmo RPM para obter a árvore de distribuição *multicast*.

2.3.1.3 *Protocol Multicast Independent* - PIM

O protocolo PIM é um dos protocolos mais utilizados para distribuição de tráfego *multicast*. Ele possui duas versões diferentes: *Dense Mode* (DM) (ADAMS; NICHOLAS; SIADAK, 2005) e *Sparse Mode* (SM) (FENNER et al., 2016). O PIM utiliza dados de

roteamento *unicast* para criação dos caminhos reversos (RPF). Além disso, ele evita criar a tabela de roteamento, utilizando informações já disponíveis nos roteadores.

A versão PIM-DM foi projetada para redes densas, onde quase todos os *hosts* são clientes de algum grupo *multicast*. Este método é eficiente quando há pelo menos um cliente ativo em cada sub-rede. Além disso, muita informação de estado é armazenada nos roteadores. O mecanismo de distribuição de pacotes é feito por árvores de distribuição baseadas em fonte.

Por outro lado, a versão PIM-SM é aplicada em um contexto onde há um pequeno grupo *multicast* constituído de diferentes redes possíveis. Ao contrário do PIM-DM, ele funciona sob demanda que parte dos *hosts* interessados. Além disso, utiliza-se árvore de distribuição compartilhada e cada fonte entrega os pacotes de dados ao RP que os encaminha para os membros do grupo *multicast*.

2.3.2 Protocolos inter-domínio

Os protocolos inter-domínio são utilizados para comunicação via *multicast* entre diferentes AS's na Internet. Assim sendo, é possível que diferentes domínios possam estabelecer comunicação entre si. Isso sem abrir mão de suas políticas de roteamento no próprio domínio.

2.3.2.1 *Multicast Source Discovery Protocol* - MSDP

O protocolo MSDP permite interconectar RP's de diferentes domínios (MCBRIDE; MEY-LOR; MEYER, 2006). Deste modo, um *host* cliente pode receber pacotes de dados de um fonte de um AS de outro domínio, considerando que a fonte envia a informação para o RP. Ele pode ser usado tanto no IPv4 quanto IPv6.

Este protocolo é utilizado em conjunto com o protocolo PIM-SM para permitir que membros de grupos remotos recebam a informação proveniente de um dado grupo *multicast*. Para tanto, em cada domínio a árvore de distribuição utilizada é uma árvore compartilhada e, portanto, deve conter o RP definido.

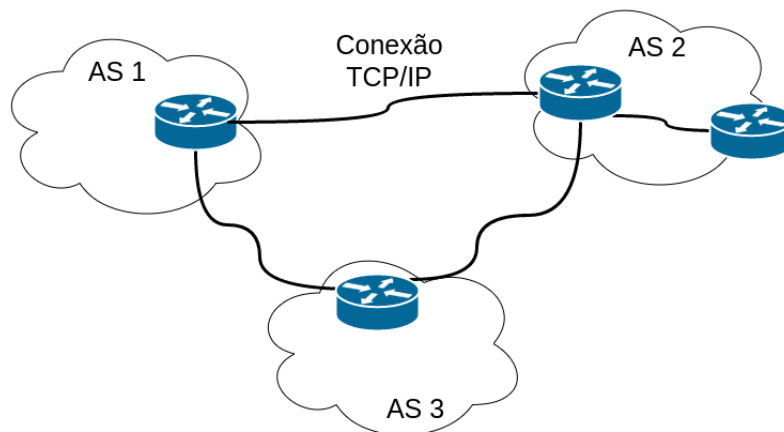
2.3.2.2 *Multiprotocol Border Gateway Protocol* - MBGP

O protocolo de roteamento MBGP é usado no contexto inter-domínio (BATES et al., 2007). Trata-se de uma extensão do protocolo *Border Gateway Protocol* (BGP) (REKH-TER; LI; HARES, 2006).

O BGP permite que RP's de diferentes AS's se comuniquem e assim o tráfego de um determinado domínio possa ser enviado para uma parte interessada que esteja em outro domínio. Para isso, alguns roteadores de cada domínio são configurados para estabelecer conexão com roteadores de outros domínios. Estes roteadores são denominados **BGP speakers**. Eles se conectam a outros *BGP speakers* em outros domínios, assim como

aos roteadores dentro do próprio domínio por meio de conexões TCP/IP. Na Figura 8, é ilustrado como ocorre a comunicação entre diferentes domínios.

Figura 8 – Comunicação entre AS's usando BGP.



Fonte: própria.

Informações sobre roteamento são trocadas entre os roteadores BGP pares conforme ilustrado na Figura 8. Além disso, todos os roteadores BGP *speakers* dentro de um domínio também devem estar em pares. Para comunicação, o protocolo troca informações utilizando o protocolo *Network Layer Reachability Information* (NLRI).

As mensagens NLRI possuem duas informações importantes para o roteamento: AS-PATH e NEXT-KEY. A informação referente ao AS-PATH é utilizada para descrever os AS's que o pacote deve passar até chegar ao destino pretendido. Já o NEXT-KEY é a informação utilizada para um roteador, digamos roteador B, informar ao próximo vizinho, digamos roteador A, qual endereço ele deve usar para encaminhar pacotes até B.

O MBGP inclui recursos não presentes no BGP. Entretanto, ele utiliza toda a política de roteamento que já existe. Além disso, ele adiciona suporte a rotas IPv6 *unicast* e IPv4 *multicast*. Este protocolo não atua diretamente na construção de árvores de distribuição - o trabalho de criação das árvores de distribuição fica a cargo do PIM.

Embora não atue diretamente na construção de árvores de distribuição, o protocolo MBGP faz um papel importante. Ele possui duas tabelas de roteamento: *Multicast Routing Information Base* (M-RIB) e *Unicast Routing Information Base* (U-RIB). As informações provenientes da tabela M-RIB permitem que o algoritmo RPF faça checagens necessárias ao protocolo PIM.

2.3.2.3 *Source Specific Multicast* - SSM

O protocolo SSM é uma extensão do protocolo PIM (HOLBROOK; CAIN, 2006), tendo como base a versão PIM-DM. Entretanto, apenas um subconjunto das configurações é

utilizado.

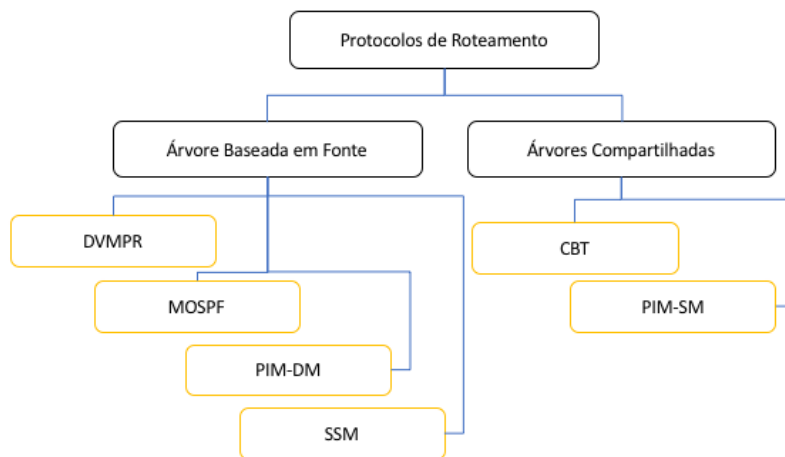
Este protocolo é utilizado para aplicações um-para-muitos onde se pode definir exatamente que fonte atenderá um determinado grupo. Ele elimina a necessidade de descoberta de roteadores RP, dispensa a criação de árvores compartilhadas e também dispensa comunicação entre RP's de domínios diferentes.

A entrega de pacotes de dados via *multicast* utiliza árvores baseadas em fonte. Além disso, o protocolo utiliza tanto IGMP quanto o MLD para controlar a entrada/saída de um membro do grupo *multicast* ao qual ele está associado.

2.3.3 Classificação dos Protocolos de Roteamento

Neste trabalho, serão discutidos apenas os algoritmos e modelos relacionados a construção de árvores baseadas em fonte. No Capítulo 3, serão apresentados modelos e algoritmos que tratam diferentes abordagens do problema de roteamento *multicast* em grafos, incluindo modelos propostos.

Figura 9 – Abordagens de construção de árvores de distribuição e protocolos.



Fonte: própria.

Na Figura 9, as abordagens para construção de árvores discutidas neste trabalho são apresentadas de forma resumida. Os protocolos podem ser agrupados de acordo com a abordagem de construção de árvores de distribuição.

2.4 Qualidade de Serviço

Inicialmente, a comunicação entre *hosts* na internet dava-se por meio de transmissão ponto-a-ponto (*unicast*). As aplicações iniciais não demandavam condições mínimas para seu funcionamento. Diante do crescimento da rede e surgimento de aplicações multimídia, tornou-se necessário desenvolver meios para atender as demandas das novas aplicações. O *multicast* tem um papel importante neste contexto, pois é uma tecnologia que visa a economia de recursos da rede.

A modelagem do problema de roteamento torna-se complexa diante das diferentes demandas associadas a uma ampla variedade de serviços multimídia (MASIP-BRUIN et al., 2006; YU; NI; ZHENG, 2011; WANG; HOU, 2000). A lista a seguir apresenta alguns exemplos de serviços.

Serviços Multimídia:

- Conferência de Áudio/Vídeo (WILLEBEEK-LEMAIR; SHAE, 1997);
- Educação a Distância (HANSEN; JAZSKIEWICS, 1998);
- Jogos interativos distribuídos (WU, 2005);
- Voz sobre IP (VOIP) (SINGH et al., 2014);
- Distribuição de Software/Conteúdo (MA; STEENKISTE, 1997).

Cada aplicação multimídia requer que determinadas condições sejam satisfeitas. Por exemplo, o custo de configuração é uma demanda comum em praticamente todos os contextos. Outras aplicações são sensíveis a *delay end-to-end* (atraso ao longo do caminho fonte/destino). Algumas aplicações são sensíveis a perda de pacotes durante a transmissão etc.

Neste contexto, qualidade de serviço (QoS) significa entregar *stream* de dados sob certas condições (métricas) que definem níveis mínimos de critérios pré-definidos para o pleno funcionamento do serviço.

Na Tabela 1, as principais demandas de QoS são apresentadas. Observa-se que alguns dos itens da tabela podem ser formulados sob diferentes formatos e interesses. Além disso, pode-se utilizá-los tanto na condição de restrição quanto na condição de função-objetivo do modelo abordado.

2.5 Resumo

No presente capítulo, foram apresentados os principais conceitos a respeito da tecnologia de transmissão via *multicast* abordando o que é mais relevante para o trabalho proposto.

Tabela 1 – Principais demandas de QoS.

Parâmetro	Descrição
Custo	Somatória das arestas utilizadas na solução.
Utilização de arestas	Representa uma demanda de utilização da capacidade disponível.
Contagem de <i>hops</i>	Número de dispositivos da rede entre fonte e destino. Longos caminhos influenciam no <i>delay</i> .
<i>Delay</i>	Corresponde ao atraso na transferência dos pacotes da fonte para os destinos. Há diferentes formulações.
<i>Jitter</i>	Diferença entre o <i>delay end-to-end</i> máximo de um caminho fonte/destino e <i>delay end-to-end</i> mínimo.
Perda de Pacotes	Consiste na avaliação do número de pacotes perdidos entre dois vértices da rede. Geralmente definido com probabilidade de perda.
Congestão	Associada a utilização das arestas. Carga total sobre uma aresta. Associado a gargalos na rede. Também definida como capacidade residual.
<i>Delay end-to-end</i>	Somatória do <i>delay</i> das arestas do vértice fonte até um vértice destino.

Inicialmente, descreveu-se os conceitos básicos. Em seguida, abordou-se os tipos de mecanismos de distribuição de dados via *multicast* e os tipos de protocolos empregados. Por fim, apresentou-se os principais conceitos relacionados a qualidade de serviço.

3 Roteamento *Multicast* em Grafos

Neste capítulo, a criação de árvores de distribuição *multicast* é modelada como um problema de roteamento em grafos, tendo como foco as funções-objetivo e restrições mais utilizadas. Na Seção 3.1, as definições básicas são apresentadas para auxiliar na modelagem matemática do problema. Na Seção 3.2, os principais trabalhos da literatura sobre o roteamento *multicast* com apenas um grupo são apresentados. Em seguida, na Seção 3.3, o problema é abordado considerando a versão multiobjetivo com apenas um grupo. Na Seção 3.4, os trabalhos sobre o problema de roteamento *multicast* com múltiplos grupos são apresentados. Por fim, os demais trabalhos relacionados são apresentados na Seção 3.5.

3.1 Roteamento *Multicast*

Nesta seção, são apresentados os principais elementos para representação do problema de roteamento em grafos e as diferentes abordagens utilizadas na representação do mesmo considerando árvores de distribuição baseadas em fonte.

3.1.1 Definição do problema

O roteamento de pacotes de dados via *multicast* reúne todos os elementos que permitem aplicação de algoritmos de otimização. Assim sendo, o problema pode ser estudado como um problema de roteamento em grafos. Considerando que os algoritmos de roteamento devem configurar uma árvore de distribuição, então é possível lançar mão de técnicas presentes na teoria dos grafos (OLIVEIRA; PARDALOS, 2005).

As árvores de distribuição podem ser facilmente modeladas como um problema de otimização em grafos, considerando que a fonte e os clientes do serviço são definidos como vértices do grafo. Além disso, os elementos intermediários do processo de comunicação (por exemplo, roteadores) também são definidos como vértices. Os *links* usados na comunicação entre os elementos do sistema *multicast* podem ser definidos como arestas. Logo, a rede onde a árvore de distribuição é configurada pode ser representada como um grafo. No restante do texto, os termos arestas e *links* denotaram uma ligação entre um par de vértices.

3.1.2 Representação em Grafos

Nesta seção, a terminologia básica e mais geral utilizada no restante do texto para revisar os modelos, descrever novos modelos e algoritmos será definida. As definições sobre grafos

são as seguintes:

- $G = (V, E)$ - grafo que representa uma rede;
- V - representa o conjunto de vértices do grafo G ;
- E - representa o conjunto de arestas do grafo G ;
- $c_{(i,j)}$ - representa o custo da aresta (i, j) ;
- $b_{(i,j)}$ - representa o capacidade de tráfego da aresta (i, j) .

As informações ilustradas na lista acima são suficientes para definir a estrutura básica de uma topologia de rede. Informações adicionais e específicas de modelo/ algoritmo serão definidas no momento que for mais apropriado para entendimento. A fim de representar as informações dos grupos *multicast*, a lista seguir define os principais elementos:

- W - representa o conjunto de grupos;
- $w_k \in W$ - representa o k -ésimo grupo;
- S_k - representa o conjunto de vértices fontes do k -ésimo grupo;
- D^k - representa o conjunto de vértices clientes do k -ésimo grupo;
- t^k - representa demanda de tráfego associada a k -ésima árvore;
- T^k - representa a k -ésima árvore.

A lista definida acima apresenta os principais elementos para definição de modelos/algoritmos considerando informações do *multicast*. Observe-se, que um grupo *multicast* pode ser representado como: $w_k = \{S_k, D^k, t^k\}$.

Conforme será detalhado adiante, a cardinalidade dos conjuntos W e S_k determinam a característica do problema abordado.

3.1.3 Generalizações do Problema de Roteamento *Multicast*

O problema de roteamento *multicast* tem sido abordado sob diferentes perspectivas de modelagem. Originalmente, o problema foi abordado considerando apenas um grupo e a configuração de uma árvore de distribuição cujo objetivo foi otimizar o custo da solução (KOMPELLA; PASQUALE; POLYZOS, 1993; WAXMAN, 1988). No entanto, rapidamente, diferentes abordagens surgiram adicionando novas demandas aos modelos inicialmente propostos.

Na lista a seguir são ilustradas, resumidamente, algumas abordagens diretamente relacionadas a este trabalho:

- **Roteamento *Multicast* Mono-Sessão**($|W| = 1$ e $|S_k| = 1$) - apenas um grupo precisa estabelecer a comunicação com a fonte que distribui a informação. Assim sendo, apenas uma árvore de distribuição é configurada. O problema é abordado com apenas uma função-objetivo.
- **Roteamento *Multicast* Mono-Sessão Multiobjetivo**($|W| = 1$ e $|S_k| = 1$) - apenas um grupo será configurado, mas o problema é abordado com mais de uma função-objetivo (ROY; BANERJEE; DAS, 2002).
- **Roteamento *Multicast* Multissessão**($|W| > 1$ e $|S_k| = 1$) - há mais de um grupo e, portanto, há necessidade de mais de uma árvore de distribuição (CHEN; GUNLUK; YENER, 2000; ZHENG; CHO; XIA, 2015).
- **Roteamento *Multicast* Multissessão Multi-Fonte**($|W| > 1$ e $|S_k| > 1$) - há mais de um grupo e cada grupo pode ser atendido por uma ou mais fontes (CHEN et al., 2013).

A configuração de uma árvore de distribuição conectando o vértice fonte aos vértices clientes pode ser visto como uma generalização do problema da árvore de Steiner. O problema da árvore de Steiner consiste em conectar um conjunto de vértices denominados terminais através de uma árvore que tenha custo mínimo (HWANG; RICHARDS; WINTER, 1992).

Assim sendo, pode-se considerar que o conjunto de vértices formados pela união do vértice fonte com os vértices clientes seja interpretado como um conjunto de vértices terminais. Esta interpretação permite assumir que o problema de roteamento *multicast* é uma generalização do problema da árvore de Steiner (OLIVEIRA; PARDALOS, 2005).

O problema da árvore de Steiner é um problema NP-Difícil (GAREY; JOHNSON, 1979). Logo, pode-se considerar que o problema de roteamento *multicast* é também NP-Difícil (OLIVEIRA; PARDALOS, 2005). As demais abordagens para o problema do roteamento *multicast* considerando mais de um grupo são também considerados NP-Difícil, em Chen et al. (2013) uma prova de tal afirmação é apresentada.

3.2 Problema de Roteamento *Multicast* Mono-Sessão

Esta versão do problema de roteamento *multicast* em grafos é a mais simples e amplamente estudada. Isso se deve ao fato de haver necessidade de configurar apenas uma árvore de distribuição. Além disso, apenas uma fonte é utilizada para atender aos membros do grupo.

3.2.1 Abordagens Mono-Fonte

Noronha e Tobagi (1994) apresentam um modelo de programação inteira geral para o problema de roteamento *multicast*. Esta formulação considera que cada aresta possui custo e *delay*. O modelo considera a otimização de uma função-objetivo que alia o valor de custo e *delay* da árvore construída. A função-objetivo utiliza uma ponderação dos valores de custo e *delay end-to-end* (*delay* acumulado ao longo do caminho fonte/cliente), que deve ser minimizado. Além disso, o modelo considera a possibilidade de otimização de mais de uma árvore. O limite de capacidade das arestas é tratado como restrição.

Em (WANG; SHI; LI, 2009), o problema de roteamento *multicast* é abordado considerando duas restrições relacionadas a *delay end-to-end*. A primeira restrição considera o ***delay end-to-end*** ao longo de um caminho entre fonte e vértice de destino (cliente). O caminho com maior *delay end-to-end* entre todos os caminhos que ligam o vértice fonte a cada um dos membros do grupo deve estar abaixo de um limite dado. A segunda restrição é denominada ***delay variation*** que consiste na diferença entre o *delay end-to-end* máximo e o *delay end-to-end* mínimo entre os caminhos da fonte para cada um dos membros do grupo. O valor obtido deve estar abaixo de um limite dado.

Wang et al. (2011) abordam o problema de roteamento *multicast* considerando quatro restrições. A restrição de *delay end-to-end* do caminho da fonte para um membro do grupo. Além dessa restrição, considera-se a **perda de pacote** em uma aresta. Esta restrição considera o produto da perda de pacotes ao longo do caminho entre fonte e membro de um grupo. Para isso, cada aresta possui uma taxa de perda de pacotes. Outra restrição utilizada tem relação com a existência de gargalo na rede - **congestionamento**, que deve ser minimizado. Considera-se a aresta mais utilizada entre todos os caminhos de fonte para cada membro do grupo. Por fim, há também a consideração do ***jitter***¹ avaliado em função do *delay* de um caminho específico e do *delay* médio entre todos os caminhos da fonte para cada membro do grupo.

Sun et al. (2008) abordam o problema considerando otimização do custo e múltiplas restrições. As restrições utilizadas são *delay end-to-end* e congestionamento. Além disso, define-se uma restrição relacionada a utilização das arestas - denominada de **utilização de banda**. Esta restrição é definida como resultado da soma do tráfego já existente na rede e uma demanda de tráfego, que é dividida pela capacidade das arestas. As três restrições são definidas com valores limites que devem ser respeitados para garantir a viabilidade da solução.

Em geral, uma ampla maioria dos trabalhos abordam o problema de roteamento *multicast* Mono-Sessão considerando a otimização do custo sujeito a restrição de *delay end-to-end*. Por vezes, consideram-se outras restrições como mencionado anteriormente. Uma boa revisão é apresentada em (XU, 2011) e vários modelos são avaliados em (XU; QU,

¹ Medida que avalia a variação de *delay* em um período de tempo.

2012; PATEL; KABAT; TRIPATHY, 2014). Muitas restrições são tratadas em alguns trabalhos como função-objetivo como é discutido na Seção 3.3.

3.2.2 Abordagem Multi-Fonte

O trabalho realizado por Guo et al. (2017) considera o problema de criação de árvore de distribuição *multicast* em um cenário que possibilita o uso de mais de uma fonte em atendimento a um conjunto de clientes. Apenas uma sessão *multicast* é considerada. Os autores sustentam que o aumento no número de fontes utilizadas diminui o custo de configuração da solução. Isso se comprova nos experimentos. Dois algoritmos foram propostos para resolução do problema: o P-MCF e o E-MCF.

A ideia do algoritmo P-MCF é criar um grafo completo G' a partir da união dos vértices fonte S_k e os vértices D^k , formando o conjunto $V' = S_k \cup D^k$. As arestas do grafo G' são caminhos entre pares de vértices de V' no grafo original, G . Em seguida, a árvore geradora T é obtida no grafo G' . Por fim, algumas arestas são removidas de T de modo que os vértices $s \in S_k$ não estejam conectados por nenhum caminho. O critério de remoção é baixar o custo final da solução. Uma floresta será retornada pelo procedimento com um árvore para cada fonte.

O segundo método, E-MCF, é uma melhoria do P-MCF. A modificação consiste em incluir vértices x não-fonte e não-cliente ao conjunto V' . Estes vértices são denominados *vértices compartilhados*. Eles são obtidos durante a fase de construção do grafo G' . Todos os vértices não-fonte e não-cliente são agrupados em um conjunto X . Em seguida, verifica-se o número de vizinhos que cada vértice tem dentro do conjunto X . Caso o valor seja maior ou igual a 3, o vértice é considerado compartilhado. Os demais são removidos do conjunto X . O novo conjunto $V' = S_k \cup D^k \cup X$ é formado. O restante do procedimento é igual ao P-MCF. Os resultados mostram que tal modificação melhorou o procedimento original.

3.3 Problema de Roteamento

Multicast Mono-Sessão Multiobjetivo

Uma variação importante do problema de roteamento *multicast* Mono-Sessão consiste na utilização de múltiplas funções-objetivo. Inúmeros trabalhos foram realizados e diferentes formulações apresentadas com variado número de funções-objetivo. Nesta seção, são apresentados os trabalhos mais relevantes e também as abordagens utilizadas para modelagem com foco nas funções-objetivo e restrições.

3.3.1 Abordagens

O trabalho pioneiro de Roy, Banerjee e Das (2002) iniciou a busca por modelos e algoritmos para tratar o problema de roteamento *multicast* sob uma visão multiobjetivo. Eles utilizaram três funções-objetivos: *end-to-end delay*; congestionamento; uma função para garantia de banda. O algoritmo proposto é uma adaptação do Non-Dominated Sorting Algorithm II (NSGA2) (ZITZLER et al., 2002). A população é ranqueada com base no critério de não-dominância. O algoritmo utiliza torneio binário para selecionar os indivíduos para aplicação dos operadores de reprodução. Os resultados foram melhorados por (CUI; LIN; WEI, 2003) através de um algoritmo evolucionário baseado na dominância de Pareto.

Crichigno e Barán (2004a) apresentam uma formulação com 4 objetivos. Além da otimização do custo da árvore de distribuição, otimiza-se as seguintes funções-objetivos: **delay médio** que consiste da somatória do *delay* sobre os caminhos da fonte para cada membro do grupo dividido pelo número de membros; *delay end-to-end* e congestionamento. O algoritmo proposto é baseado na abordagem Strenght Pareto Evolutinoary Algorithm (SPEA), que utiliza uma tabela de roteamento contendo caminhos entre o vértice fonte e os vértices membros do grupo. Estes caminhos são criados através do algoritmo de Yen (1971) que produz k-caminhos mais curtos. São armazenados caminhos considerando três critérios: caminho mais barato, caminho mais curto e caminho com arestas menos utilizadas. Em (CRICHIGNO; BARÁN, 2004b), apresentam-se dois casos de teste e os respectivos conjuntos de Pareto ótimo.

Fabregat et al. (2004) abordam o problema de roteamento *multicast* considerando 4 funções-objetivo relacionadas a qualidade de serviço. O conjunto de soluções não-dominadas é obtido considerando a minimização do **número de hops** entre fonte e os membros do grupo, minimização da congestão da rede (definida pela aresta mais congestionada), minimização do *delay end-to-end* e maximização da utilização da banda. Além disso, o modelo inclui uma restrição associada ao limite de capacidade das arestas. Um algoritmo genético é aplicado. A população inicial é criada utilizando um algoritmo de busca em profundidade. Os caminhos entre fonte e cada membro do grupo são codificados em um cromossomo de tamanho igual ao número de caminhos. Em seguida, as soluções são colocadas em um conjunto de soluções não-dominadas. Operadores de cruzamento e mutação são aplicados.

Pinto, Barán e Fabregat (2005) propuseram um modelo otimizando três funções-objetivo. A otimização considera o custo de configuração da árvore de distribuição, *delay end-to-end* da fonte para cada membro do grupo (minimiza o *delay* máximo) e o *delay* médio que consiste da somatório do *delay* de cada caminho da árvore dividido pelo número de membros (número de caminhos). O autores desenvolveram um algoritmo baseado em colônia de formigas. Adicionalmente, criou-se um processo de construção de árvores colocando uma formiga no vértice fonte e deixando percorrer a rede até montar a árvore.

O algoritmo constrói m soluções por iteração.

Fabregat et al. (2005) desenvolveram um importante trabalho de revisão do problema de roteamento *multicast* multiobjetivo. O trabalho considera a análise de inúmeros trabalhos até aquele momento. Apresenta-se um resumo dos principais elementos utilizados como funções-objetivo e as restrições dos modelos analisados. Resumiu-se 11 funções-objetivos relacionadas a diferentes interesses de otimização e 4 restrições. Além disso, um modelo matemático geral é proposto com 11 funções-objetivos e 21 restrições diferentes. Os autores também apresentam um estudo algorítmico utilizando a abordagem SPEA2.

Huang e Liu (2010) abordam o problema de roteamento *multicast* em redes móveis *ad-hoc*. O modelo utilizado possui quatro objetivos relacionados a qualidade de serviço. O modelo considera congestão da rede como sendo a aresta com menor capacidade disponível, que deve ser maximizada. Além disso, otimiza-se a perda de pacotes, *delay end-to-end* e o *jitter* definido como a diferença entre o *delay end-to-end* máximo e o *delay end-to-end* mínimo - a maior diferença deve ser minimizada. Os autores apresentam um algoritmo genético para resolução com operadores de mutação e *crossover*.

Ning, Zhang e Zhang (2016) desenvolveram um modelo com três funções-objetivo. Os autores otimizam o *delay end-to-end*, o custo de configuração e a largura de banda (definida como a aresta com menor capacidade disponível na árvore de distribuição). Este último objetivo é o mesmo que minimizar a congestão. Um algoritmo baseado em colônia de abelhas artificiais é proposto. Uma representação por matriz é utilizada onde cada linha representa um caminho entre fonte e um membro do grupo. A criação de soluções é baseada em busca em profundidade e árvore geradora mínima.

Em Xu (2011), um extenso estudo é realizado abordando o problema de roteamento *multicast* multiobjetivo. O modelo proposto e utilizado considera cinco funções-objetivo. Os objetivos de otimização são minimizar o custo, minimizar o *delay end-to-end*, minimizar a utilização de banda, minimizar o *delay* médio e minimizar a variação de *delay* entre os caminhos da fonte para cada membro do grupo. Este trabalho apresenta uma boa revisão do estado da arte até o momento de sua finalização. Também são discutidas várias abordagens algorítmicas para o problema.

3.4 Roteamento *Multicast* Multissessão

Nesta seção, são discutidos algoritmos e modelos dos trabalhos diretamente relacionados a esta proposta. Considera-se o problema de roteamento *multicast* com múltiplos grupos e, portanto, mais de uma sessão ativa na rede - uma por grupo.

Cada grupo tem uma árvore de distribuição associada a ele. deve ao fato de utilizar-se árvores de distribuição baseadas em fonte. Logo, haverá concorrência por recursos da rede - por exemplo capacidade das arestas.

3.4.1 Congestão

Nesta versão do problema, os modelos e algoritmos focam na otimização da congestão na rede. Em termos práticos, procura-se melhorar a distribuição do uso da capacidade das arestas da rede a fim de evitar gargalos. Os modelos não consideram o valor de capacidade $b_{(i,j)}$ de cada aresta. Assume-se que cada grupo demanda um valor máximo, t^k , de capacidade da aresta quando a mesma for utilizada em uma árvore de distribuição.

Definição 1. *Congestão da rede é definida em função da capacidade das arestas, especificamente a aresta com o menor valor de capacidade disponível para utilização determina esse valor.*

Considerando a Definição 1, Chen, Gunluk e Yener (2000) propõem dois modelos que visam a otimização do valor de congestão. Especificamente, busca-se a minimização da congestão máxima, que é representada pela aresta com maior carga de tráfego.

Os modelos ilustrados a seguir utilizam elementos definidos na Seção 3.1.2 e, adicionalmente os seguintes elementos:

λ - representa o valor de congestão máximo;

$x_{(i,j)}^k \in \{0, 1\}$ - variável binária que indica se a aresta (i, j) faz parte da árvore T^k .

O cálculo do valor de λ não é mencionado no trabalho dos autores, entretanto é possível defini-lo considerando $t^k \geq 1$ conforme a Equação 3.1:

$$\lambda = \sum_{(i,j) \in E} t^k \cdot x_{(i,j)}^k, \forall k \in W. \quad (3.1)$$

O primeiro modelo proposto pelos autores minimiza o valor de congestão. A Equação 3.2 induz uma árvore e a Equação 3.3 indica que o tráfego gerado sobre as arestas é menor ou igual o valor máximo de congestão.

Minimizar λ

Sujeito à :

$$x^k \in T^k = \{x^k \in \{0, 1\}^{|E|} \text{ gera uma árvore}\}, \forall k \in W \quad (3.2)$$

$$\sum_{k \in W} t^k x_{(i,j)}^k \leq \lambda, \forall (i, j) \in E \quad (3.3)$$

O modelo proposto não considera o uso de uma restrição de capacidade nas arestas, restrição que é amplamente usada em problemas de roteamento *multicast*.

O segundo modelo incorpora o custo de configuração da solução por meio de uma penalidade na função-objetivo. A ideia é limitar o crescimento da árvore, considerando

que minimizar a congestão máxima pode levar ao crescimento dos caminhos entre fonte e os membros de cada grupo. Para tanto, considere-se Equação 3.4:

$$OPT^k = \sum_{(i,j) \in E} c_{(i,j)} \cdot x_{(i,j)}^k \quad (3.4)$$

A Equação 3.4 determina o valor de OPT^k . Este valor representa custo ótimo da k -ésima árvore de distribuição obtido isoladamente das demais árvores. Se o valor de $c_{(i,j)}$ é igual a 1 para todas as arestas, então se tem o valor ótimo com relação ao tamanho da árvore (tamanho medido como número de arestas).

A Restrição 3.5 está associada a um variável que controla a dilatação² da k -ésima árvore de distribuição, $\pi^k \geq 0$. Seu valor deve ser sempre maior que o *gap* entre o valor ótimo OPT^k e o valor de custo $c_{(i,j)} \cdot x_{(i,j)}^k$. Assim, há um controle no crescimento do custo ($c_{(i,j)} \geq 1$) ou no tamanho da árvore ($c_{(i,j)} = 1$). Os autores utilizaram $\alpha = 2$.

$$\pi^k \geq \frac{1}{OPT^k} \left\{ \sum_{(i,j) \in E} c_{(i,j)} \cdot x_{(i,j)}^k - \alpha OPT^k \right\} \quad (3.5)$$

A inclusão da penalidade na função-objetivo dá-se por meio da adição de um parâmetro de penalidade p^k . A Equação 3.6 ilustra o cálculo da penalidade. Trata-se de somar o valor da variável π^k vezes o parâmetro p^k para cada grupo $k \in W$.

$$\sum_{k \in W} p^k \pi^k \quad (3.6)$$

Com a definição da restrição de dilatação na Equação 3.5 e da penalidade de custo na Equação 3.6, o segundo modelo proposto é definido a seguir:

$$\text{Minimizar } \lambda + \sum_{k \in W} p^k \pi^k$$

Sujeito à :

$$x^k \in T^k = \{x^k \in \{0, 1\}^{|E|} \text{ gera uma árvore}\}, \forall k \in W$$

$$\sum_{k \in W} t^k \cdot x_{(i,j)}^k \leq \lambda, \forall (i, j) \in E$$

$$\pi^k \geq \frac{1}{OPT^k} \left\{ \sum_{(i,j) \in E} c_{(i,j)} \cdot x_{(i,j)}^k - \alpha \cdot OPT^k \right\}$$

Esta abordagem é pouco difundida na literatura. Outros trabalhos utilizaram um visão diferente para tratar do congestionamento das arestas da rede. Por exemplo, utilização da otimização do valor de capacidade residual, que será discutido na Seção 3.4.2.

² Dilatação é o aumento no número de arestas, comprimento do caminho da árvore.

Algoritmo

Os autores propuseram um algoritmo para resolução do modelo. O algoritmo proposto precisa de uma etapa de pré-processamento. Esta etapa é necessária para obter o valor OPT^k para o k -ésimo grupo. Para tanto, utilizou-se um algoritmo baseado na abordagem *branch-and-bound* (KOCK; MARTIN, 1998) para obter o valor de cada árvore de distribuição.

Ao final do pré-processamento, um conjunto de árvores com valor de custo ótimo é obtido - ressalte-se que o valor OPT^k pode representar o número de arestas. O primeiro passo é avaliar o valor de congestão máxima referente ao conjunto de árvores e em seguida, um processo de troca de arestas é iniciado para remover as arestas mais congestionadas.

3.4.2 Capacidade Residual

Nesta abordagem, o objetivo é otimizar a capacidade residual da rede. Este objetivo está relacionado ao congestionamento proveniente do uso das arestas durante a ativação das sessões *multicast*. Este problema foi abordado em (LEE; CHO, 2004), (YOUSSEF et al., 2002) e (ANDRADE; GOLDBARG; GOLDBARG, 2015).

3.4.2.1 Capacidade Residual Sujeito a Limite de Crescimento das Árvores

Lee e Cho (2004) propuseram a otimização da capacidade residual (Definição 2) sujeito a limitação no crescimento das árvores de distribuição. Este trabalho foi comparado contra os resultados obtidos por Chen, Gunluk e Yener (2000) e se mostrou melhor tanto na resolução quanto na modelagem do problema. A consideração do limite de capacidade das arestas é um fator importante da formulação, pois isto aproxima o modelo do problema real.

Os elementos definidos a seguir em conjunto com os já definidos na Seção 3.1.2 serão utilizados para apresentar o modelo dos autores:

$d(S)$ - arestas com um vértice pertencente a $V \setminus S$ e outro vértice pertencente a S .

V representa vértices da rede;

v_i^k - indica se o vértice i está na árvore T^k ;

$x_{(i,j)}^k$ - variável binária indicando se a aresta (i, j) está na árvore T^k ;

$E(i)$ - representa as arestas que contém o vértices i .

Definição 2. *Capacidade Residual* - o valor de capacidade da aresta menos o valor de capacidade já utilizado. O valor é calculado conforme a Equação 3.7.

A Equação 3.7 define o valor de capacidade residual de uma aresta:

$$Z_{ij} = b_{(ij)} - \sum_{k \in W} x_{(i,j)}^k \cdot t^k, \quad \forall (i, j) \in E \quad (3.7)$$

A partir da definição do conceito de capacidade residual e da definição do cálculo para uma aresta (i, j) , define-se o valor de capacidade residual mínima na Equação 3.8:

$$Z = \min_{(i,j) \in E} Z_{ij} \quad \forall (i, j) \in E \quad (3.8)$$

O modelo proposto por Lee e Cho (2004) é apresentado nas Equações 3.9-3.16 e tem como função-objetivo a Equação 3.8. Os conjuntos de Restrições 3.9, 3.10 e 3.11 indicam que o vértice i está conectado a árvore T^k por uma aresta (i, j) , que pelo menos uma aresta pertencente ao conjunto $E(i)$ está presente em T^k e que o número de arestas em T^k somado de 1 é igual ao número de vértices em T^k , respectivamente. A Restrição 3.13 garante que a capacidade residual será maior ou igual ao valor de Z .

Maximizar Z

Sujeito à:

$$\sum_{(i,j) \in d(S)} x_{(i,j)}^k \geq 1 \quad \forall k \in W \quad \forall S \subset V \mid S \cap D^k \neq \emptyset \quad (3.9)$$

$$v_i^k \geq x_{(i,j)}^k \quad \forall v_i^k \quad \forall k \in W \quad \forall (i, j) \in E(i) \quad (3.10)$$

$$v_i^k \leq \sum_{(i,j) \in E(i)} x_{(i,j)}^k \quad \forall i \in V \quad \forall k \in W \quad (3.11)$$

$$\sum_{i \in V} v_i^k = 1 + \sum_{(i,j) \in E} x_{(i,j)}^k \quad \forall k \in W \quad (3.12)$$

$$b_{(i,j)} - \sum_{k \in W} t^k x_{(i,j)}^k \geq Z \quad \forall (i, j) \in E \quad (3.13)$$

$$\sum_{(i,j) \in E} x_{(i,j)}^k \leq \alpha OPT^k \quad \forall k \in W \quad (3.14)$$

$$z \geq 0 \quad (3.15)$$

$$v_i^k, x_{(i,j)}^k \in \{0, 1\} \quad (3.16)$$

A Restrição 3.14 é utilizada para limitar o crescimento da árvore a medida que se otimiza o valor de Z . Quando Z é maximizado, naturalmente a carga de tráfego sobre as arestas é distribuída entre arestas com maior capacidade residual. Deste modo, é possível elevar o valor de Z . Isto se deve ao fato do valor baixo de Z representar alto consumo da capacidade de um conjunto pequeno de arestas. Então, distribuir este consumo para outras arestas pode levar ao crescimento das árvores. Nos experimentos realizados o valor $\alpha = 2$ foi utilizado.

Algoritmos

Lee e Cho (2004) propuseram dois algoritmos baseados em substituição de arestas. Os algoritmos consideram dois cenários distintos para lidar com requerimento de tráfego

uniforme e igual a 1 ($t^k = 1$) e requerimento de tráfego não-uniforme ($t^k \geq 1$).

O primeiro algoritmo denominado *Highest bandwidth-first* é aplicado ao cenário com $t^k = 1$. Inicialmente, as árvores de distribuição referentes ao k -ésimo grupo são criadas. Em seguida, avalia-se a capacidade residual das arestas pertencentes a solução. Após esta avaliação, as arestas são ordenadas de modo crescente por valor de capacidade residual. Neste momento, há a tentativa de substituí-las através da remoção da aresta mais congestionada e reconexão da árvore com algoritmo de caminho mais curto. As arestas que entrarão na árvore após a substituição devem conter capacidade residual $Z_{ij} = Z + 2$. Assim sendo, tem-se o ganho de 1 unidade de capacidade residual.

O segundo algoritmo proposto é denominado *Closest-Gap-First*. Ele é aplicado ao cenário onde $t^k \geq 1$. Os mesmos procedimentos de construção de árvores de distribuição e avaliação da capacidade residual mencionados no parágrafo anterior são aplicados. No entanto, a estratégia de escolha da aresta que vai entrar na árvore para melhorar a capacidade residual é diferente. Os autores definiram o termo *alternative-gain* para indicar a seguinte situação: considere Z^1 e Z^2 a primeira e segunda arestas com menor capacidade residual, respectivamente. Então, a remoção da aresta de capacidade Z^1 leva a um ganho de no máximo $Z^2 - Z^1$. As arestas com valor de capacidade mais próximo dessa diferença são escolhidas.

3.4.2.2 Capacidade Residual Sujeito com Suporte para Múltiplas Fontes

O modelo proposto por Chen et al. (2013) segue na mesma linha de pensamento do modelo definido em (LEE; CHO, 2004). Entretanto, o novo modelo não considera um limite de tamanho para o crescimento das árvores, apenas se utiliza uma restrição de capacidade das arestas. O modelo incorpora um recurso interessante que é a capacidade de lidar com múltiplas fontes por grupo.

Além das definições apresentadas na Seção 3.1.2, considere as seguintes:

$x_{(i,j)}^k$ - variável binária que representa o fluxo que passa pela aresta (i, j) em caso de tratar o problema como um problema de fluxo;

$F_{(i,j)}^k$ - é um variável binária que assume valor 1 quando $x_{(i,j)}^k$ for positivo e 0 caso contrário

$A_{(v)}$ - representa os vizinhos de vértice $v \in V$;

$c = \max_{k \in W} \{|D^k|\}$ - parâmetro utilizado no modelo.

O conjunto de Restrições 3.17-3.25 definem o modelo matemático para otimização da capacidade residual considerando múltiplas fontes por grupo. As Restrições 3.19-3.21 são Restrições comuns em problemas de roteamento utilizando fluxo: as Restrições 3.19 garantem que o fluxo que sai do conjunto de fontes S_k é igual ao fluxo que chega ao

conjunto de demanda D^k ; as Restrições 3.20 garantem que o número de cópias chegando ao vértice de demanda seja 1 a mais que o número de cópias que saem deste vértice e as Restrições 3.21 garantem a conservação de fluxo em vértices intermediários. As Restrições 3.22 são usadas para marcar arestas utilizadas para entrega dos *streams* aos vértices de demanda, onde a variável $F_{(i,j)} = 1$ quando há fluxo passando por uma aresta $(i, j) \in E$. As Restrições 3.23 garantem que não haverá fluxo entrando em nenhum vértice fonte e as Restrições 3.24 evitam que um vértice seja alcançado por mais de um vértice vizinho. Ambas as restrições evitam que ciclos sejam criados. As Restrições 3.25 garantem que a capacidade das arestas seja maior ou igual ao valor de Z .

$$\text{maximizar } Z \quad (3.17)$$

$$\text{Sujeito à:} \quad (3.18)$$

$$\sum_{i \in S_k} \sum_{j \in A(i)} X_{(i,j)}^k = |D^k|, \quad \forall w_k \in W \quad (3.19)$$

$$\sum_{j \in A(i)} X_{(j,i)}^k - 1 = \sum_{j \in A(i)} X_{(i,j)}^k, \quad \forall w_k \in W, i \in D^k \quad (3.20)$$

$$\sum_{j \in A(i)} X_{(i,j)}^k = \sum_{j \in A(i)} X_{(i,j)}^k, \quad \begin{matrix} \forall w_k \in W \\ i \in D^k \\ i \notin S_k, \\ i \notin D^k \end{matrix} \quad (3.21)$$

$$X_{(i,j)}^k \leq c F_{(i,j)}, \quad \forall i \in V, j \in A(i) \quad (3.22)$$

$$\sum_{j \in A(i)} F_{(i,j)}^k = 0, \quad \forall w_k \in W, i \in S_k \quad (3.23)$$

$$\sum_{j \in A(i)} F_{(i,j)}^k \leq 1, \quad \forall w_k \in W, i \in V, i \notin S_k \quad (3.24)$$

$$b_{(i,j)} - \sum_{w_k \in W} t^k \cdot [(F_{(i,j)} + F_{(j,i)})] \geq Z \quad \forall (i, j) \in E \quad (3.25)$$

O modelo proposto por Chen et al. (2013) permite tratar o problema de roteamento *multicast* com apenas uma fonte por grupo, assim como é possível utilizar múltiplas fontes. Em se tratando de utilizar $|S_k| = 1$, o problema é similar a versão proposta em (LEE; CHO, 2004). Entretanto, o modelo torna-se importante pela capacidade de lidar com múltiplas fontes - quando $|S_k| \geq 1$. Ademais, o trabalho pode ser ampliado e analisado de acordo com a otimização do custo de configuração da solução incorporando elementos relacionados a qualidade de serviço.

Os autores definem duas condições importantes para o atendimento de um grupo por múltiplas fontes:

- Cada membro de um grupo só pode ser atendido por uma fonte;
- As árvores de distribuição obtidas são disjuntas em vértices e arestas.

Algoritmos

Chen et al. (2013) propuseram um algoritmo heurístico capaz de resolver o problema de roteamento *multicast* otimizando a capacidade residual tanto com uma fonte quanto com múltiplas fontes. A ideia básica é construir árvores de caminho aumentado utilizando uma versão modificada do algoritmo de Dijkstra (1959).

O algoritmo de construção de árvores para um dado grupo k possui duas fases. Na primeira fase, uma árvore de caminho aumentado é criada para cada vértice fonte $s \in S_k$. Em seguida, utiliza-se um procedimento de construção das árvores finais da solução. Este processo consiste em fazer o percurso do vértice cliente até um vértice fonte. Quando o vértice cliente x acha o vértice fonte s , então significa que x está na árvore de s . A busca permite que um vértice cliente x conecte-se a s pelo melhor caminho possível (em termos de capacidade disponível nas arestas).

3.4.3 Custo de Configuração

Os modelos que visam à otimização de custo de configuração das árvores de distribuição são amplamente estudados. Eles apresentam características diferentes quanto a formulação de restrições. Entretanto, a restrição que limita a capacidade das arestas está presente em todos os modelos analisados. Ademais, há também restrições visando aspectos de qualidade de serviço como é o caso das restrições que limitam o *delay end-to-end* e o número de saltos (*hops*) no caminho entre fonte e clientes. Nesta seção, os modelos e algoritmos de otimização de custo são discutidos.

3.4.3.1 Otimização do Custo com Limite de Capacidade das Arestas

Estes modelos representam os trabalhos iniciais a respeito da otimização do roteamento *multicast* com mais de um grupo. A modelagem desta versão do problema considera apenas duas restrições: o limite de capacidade das arestas e o *delay end-to-end*.

Considere as notações da Seção 3.1.2 e, adicionalmente, as seguintes:

$E(T^k)$: arestas pertencentes a árvore T^k ;

$x_{(i,j)}^k$: variável de decisão indicando se a aresta (i, j) está na árvore T^k .

Os conjuntos de Equações 3.26 e 3.27 definem o modelo matemático geral para otimização do custo. A Equação 3.26 define a avaliação do custo da solução que deve ser minimizado. Observa-se que o custo da aresta é somado por cada participação dela em uma árvore. A Restrição 3.27 é utilizada para garantir que o limite de capacidade das arestas seja respeitado.

$$\text{Minimizar } \sum_{k=1}^{|W|} \sum_{(i,j) \in T^k} c_{(i,j)}, \quad (i,j) \in E(T^k) \quad (3.26)$$

sujeito à:

$$\sum_{k=1}^{|W|} t^k \cdot x_{(i,j)}^k \leq b_{(i,j)} \quad \text{onde} \quad x_{(i,j)}^k = \begin{cases} 1 & (i,j) \in E(T^k) \\ 0 & \text{caso contrário} \end{cases} \quad (3.27)$$

Algoritmos

Jia e Wang (1997) propuseram um algoritmo baseado em redes de distância (KOU; MARKOWSKY; BERMAN, 1981). A ideia é construir um grafo G' a partir de G . Cada aresta de G' é um caminho entre dois vértices do conjunto $C = s_k \cup D^k$. Em seguida, o algoritmo de árvore geradora mínima de Kruskal (1956) é aplicado sobre o grafo G' (KRUSKAL, 1956). Com a árvore geradora mínima configurada, as arestas obtidas (caminhos) são utilizadas para construir a árvore de distribuição. Os autores definem o conceito de *alternative overhead* que está associado ao congestionamento das arestas. Isto é aplicado durante a construção das árvores observando o estado da rede após a construção das árvores anteriores. Considere que a k -ésima árvore de distribuição está sendo construída e uma aresta (i, j) deve ser utilizada por esta árvore. No entanto, a aresta em questão está congestionada e não tem capacidade suficiente. Portanto, o *alternative overhead* de todas as árvores já construídas é calculado. A árvore que tiver menor *alternative overhead* deve ceder a aresta. Assim, a piora no congestionamento é reduzida. Em termos práticos, a árvore que ceder uma aresta sua causando o menor aumento no consumo da capacidade das arestas é escolhida.

Low e Wang (1999) apresentaram um algoritmo baseado na heurística de Takahashi e Matsuyama (1980). O algoritmo foi comparado com a proposta de Jia e Wang (1997) e obteve melhores resultados. O conceito de *alternative overhead* também é utilizado. A ideia central do algoritmo é construir uma árvore T^k para o k -ésimo grupo e, em seguida, verificar a situação das arestas quanto ao congestionamento. Se houver uma aresta saturada (capacidade esgotou-se), então T^k é reparada através da substituição dessa aresta. Computa-se o *alternative overhead* para todas as árvores já construídas e compara-se com a nova árvore. A aresta com menor *alternative overhead* deve ceder a aresta.

Wang, Liang e Jan (2002) propuseram três heurísticas. A primeira heurística é uma abordagem gulosa baseada na heurística de Takahashi e Matsuyama (1980). A cada árvore construída, o valor de capacidade das arestas é atualizado visando ao controle de arestas saturadas. Este processo é repetido até construir todas as árvores. A segunda heurística também utiliza a heurística de Takahashi. Entretanto, aplica-se a heurística para todas as árvores. Ao final do processo, é possível que haja aresta saturada, então aplica-se um

procedimento para substituição de arestas. Por fim, a terceira heurística está associada a conjuntos de corte. As árvores são extraídas da rede de forma sequencial.

3.4.3.2 Otimização do Custo com Limite de *Delay*

O modelo proposto por Yan-lin (2010) apresenta uma formulação que incorpora um elemento importante para qualidade de serviço. Trata-se da consideração do *delay end-to-end*, que é adicionado como uma restrição do problema. Adicionalmente, um limite máximo de *delay* é usado na construção de caminhos entre vértice fonte e clientes.

Considere as seguintes definições:

$D(i, j)$ - *delay* da aresta (i, j) ;

$P_{T^k}(i, j)$: *delay* do caminho entre qualquer par de vértices i e j na árvore do grupo k ;

Δ_k : tolerância a *delay* no caminho para fonte s_k e grupo k .

Considere, também, o modelo ilustrado nas Equações 3.26 e 3.27. A restrição de *delay* adicionada por Yan-lin (2010) é definida por:

$$\sum_{(i,j) \in P_{T^k}(d_i, d_j)} D(i, j) \leq \Delta_k \quad \forall k \in W \quad (3.28)$$

Algoritmo

Yan-lin (2010) apresenta uma heurística que utiliza o algoritmo de caminho mais curto de Dijkstra (1959). Inicialmente, todas as árvores são construídas. Em seguida, é verificado se houve alguma violação das restrições do problema. Por exemplo, alguma aresta teve sua capacidade excedida. A correção das violações é realizada por meio de um procedimento iterativo de troca de arestas.

3.4.3.3 Otimização de Custo com Limite *Hops*

O modelo proposto por Kang, Park e Park (2009) visa a otimização do custo de configuração da solução e adiciona uma restrição relacionada a qualidade de serviço. Trata-se da adição de controle sobre o comprimento do caminho entre fonte e clientes. Isto é feito através da contagem de *hops* (saltos) - definidos pelo número de arestas utilizadas. Esta modelagem está relacionada a restrição de *delay*, embora seja avaliada pela comprimento do caminho.

O modelo proposto é resolvido por meio de um esquema de geração de colunas (FULKERSON, 1971). Onde cada coluna consiste da resolução do problema da árvore de Steiner sujeito a limite de hops (VOSS, 1999).

Considere as seguintes definições:

- h_k - limite máximo de *hops* por grupo k ;
- F_k - conjunto de árvores de distribuição para o grupo k ;
- $F_k\{(i, j)\} \subseteq F_k$ - conjunto de árvores que incluem a aresta (i, j) ;
- $A\{k\}$ - conjunto de arestas pertencentes a árvore T^k ;
- x_{kf} - uma variável de decisão indicando se o grupo k usa a árvore de distribuição $T^k \in F_k$, $x_{kf} \in \{0, 1\}$;
- C_{kf} - custo de configuração da k -ésima árvore.

O modelo proposto é definido pelas Equações 3.29-3.32 definem:

$$\text{Minimizar } \sum_{k=1}^{|W|} \sum_{f \in F_k} C_{kf} \cdot x_{kf} \quad (3.29)$$

Sujeito à:

$$\sum_{f \in F_k} x_{kf} = 1, \quad \forall k = 1, \dots, |W| \quad (3.30)$$

$$\sum_{k=1}^{|W|} \sum_{f \in F_k\{(i,j)\}} t^k \cdot x_{kf} \leq b_{(i,j)}, \quad \forall (i, j) \in E \quad (3.31)$$

$$x_{kf} \in \{0, 1\} \quad \forall f \in F_k, \quad k = 1, \dots, |W| \quad (3.32)$$

Os autores utilizam um esquema de geração de colunas para resolução do problema, onde cada árvore é tratada como uma árvore de Steiner limitada por *hops*. A Equação 3.30 verifica se apenas uma árvore é utilizada para grupo e a Restrição 3.31 verifica se a capacidade de uma aresta (i, j) está dentro do limite.

Algoritmo

Os autores propuseram uma abordagem de resolução do problema baseada no método *branch-and-price-and-cut*. Eles utilizam cortes na árvore de *branch-and-bound* durante a resolução do problema. Além disso, o problema é relaxado usando técnicas de programação linear.

O programa linear obtido em cada nó da árvore de *branch-and-bound* - Z_{LP} é utilizado como um limite inferior para o valor-objetivo ótimo. Os autores propuseram uma abordagem baseada no método *branch-and-price-and-cut*. Durante a resolução do modelo, utiliza-se cortes nos nós da árvore de *branch-and-bound*. Para tanto, um problema é relaxado usando programação linear. Quando o valor de Z_{LP} é igual ao valor de avaliação de uma solução inteira viável, então isso significa que o valor ótimo foi encontrado. Em caso contrário, a busca por uma solução ótima continua através da exploração do restante da árvore *branch-and-bound*.

3.5 Outras Abordagens

Randaccio e Atzori (2007) propuseram uma abordagem bastante diferenciada. Eles abordaram o problema com foco inteiramente voltado para os aspectos de qualidade de serviço. Duas formulações foram idealizadas e estão associadas a duas funções que tratam da utilização da rede (compartilhamento das arestas) e *delay*. A primeira formulação considera o compartilhamento de arestas de forma isolada como critério a ser otimizado. A segunda formulação considera a agregação da função de compartilhamento junto com a função de *delay*. Um algoritmo genético foi projetado para resolução do problema. Os resultados foram comparados contra o trabalho de (CHEN; GUNLUK; YENER, 2000) e (WANG; LIANG; JAN, 2002). Vale ressaltar que os autores realizaram adaptações aos trabalhos comparados. Utilizou-se apenas as funções-objetivo dos trabalhos mencionados, adicionando-as a um algoritmo genético.

Em (WANG et al., 2015), uma nova abordagem é proposta. Trata-se da construção de árvores de distribuição com foco no consumo de energia. O modelo incorpora aspectos relacionados a qualidade de serviço. Isto é realizado por meio de limites máximos toleráveis de *delay* e *jitter*. Adicionalmente, os autores utilizam um modelo de probabilidade para determinar a chance de uma aresta compor uma árvore de distribuição. Uma heurística foi proposta para solucionar o problema. A ideia central é construir uma árvore de distribuição para cada membro do grupo, fazer um *merge* delas e em seguida remover os ciclos possivelmente criados e as folhas que não sejam vértices do conjunto de vértices que forma o grupo. Isso se repete para cada árvore.

No trabalho desenvolvido por Zheng, Cho e Xia (2015) há utilização de múltiplas sessões, onde cada sessão possui uma árvore de distribuição associada. Estas sessões devem permanecer ativas por longos períodos de tempo e estão presentes em redes estáveis. Os autores utilizam uma abordagem baseada no uso de geração de colunas para resolução de árvores de Steiner como subproblema. Utiliza-se funções utilitárias relacionadas ao consumo de banda da rede. O objetivo é maximizar o valor agregado das funções visando ao equilíbrio dos recursos sob diferentes interesses.

3.6 Resumo

Neste capítulo, o problema de roteamento *multicast* foi revisado sob ótica da otimização. Os modelos apresentados tratam o problema considerando diferentes formulações. Essas diferenças são caracterizadas principalmente pelas funções-objetivo e restrições empregadas na construção dos modelos. Além disso, o problema também foi caracterizado conforme se deu a formação dos grupos e o seu atendimento por uma fonte ou múltiplas fontes. Por fim, houve uma análise geral sobre os algoritmos empregados por cada um dos autores mencionados.

4 Otimização Multiobjetivo

Neste capítulo, apresentam-se os principais conceitos sobre a otimização multiobjetivo. Apenas os elementos mais importantes da teoria, que foram abordados no trabalho são revisados. Na Seção 4.1, apresentam-se as noções básicas da teoria. Na Seção 4.2, apresentam-se os processos utilizados para avaliação de conjuntos de aproximação. Na Seção 4.3, os principais algoritmos da literatura são apresentados. Por fim, na Seção 4.4, a Transgenética é apresentada.

4.1 Noções Básicas

Nesta seção, as principais definições sobre otimização multiobjetivo serão apresentadas. Em geral, problemas do mundo real possuem objetivos conflitantes. Estes conflitos representam situações de *trade-off*. Portanto, é necessário empregar esforços na busca de soluções ótimas considerando todos os objetivos importantes (DEB, 2008).

4.1.1 Problema Multiobjetivo

A definição de um problema de otimização multiobjetivo consiste de quatro elementos fundamentais: o espaço de decisão, o espaço objetivo, conjunto de funções-objetivo e uma relação binária (KNOWLES, 2002; KNOWLES; THIELE; ZITZLER, 2006).

Por exemplo, considere o problema da mochila multiobjetivo (JASZKIEWICZ, 2002). Este problema considera um número M de objetivos. Há um número o de objetos a serem inseridos na mochila. Cada objeto o possui um peso p_{oj} e um custo c_{oj} associado ao objetivo $j = \{1, \dots, M\}$. O peso da mochila para o objetivo j é P_j . Considerando a variável $x_i = \{0, 1\}$, que indica se o objeto $i = \{1, \dots, o\}$ está na mochila, o problema é formalmente definido como:

$$\text{Maximizar: } f_j(x) = \sum_{i=1}^o c_{oj} \quad (4.1)$$

$$\begin{aligned} \text{sujeto a: } \sum_{i=1}^o p_{oj} x_i &\leq P_j \\ j &= \{1, \dots, M\}, x_i \in \{0, 1\}, i = \{1, \dots, o\} \end{aligned} \quad (4.2)$$

As Equações 4.1 e 4.2 definem formalmente o problema da mochila multiobjetivo. A partir destas definições os elementos que formam um problema multiobjetivo serão definidos.

- **Espaço de decisão** (Φ): O espaço de decisão compreende todas as configurações de soluções que sejam factíveis - **vetor de decisão**. Esta solução é determinada por uma configuração associada às variáveis de decisão do problema. Uma solução $x \in \Phi$ do problema da mochila é representada por $x = \{x_1, x_2, \dots, x_o\}$.
- **Espaço objetivo** (Ψ): O espaço objetivo do problema é determinado em função do espaço de decisão Φ . Cada solução $x \in \Phi$ possui uma imagem no espaço objetivo determinada através de $f(x)$ (funções de avaliação). Uma solução $x \in \Phi$ é um ponto p no espaço objetivo, $p = \{p_1, p_2, \dots, p_i\}$, onde $p_i = f_i(x)$. Também denominada de **vetor-objetivo**.
- **Funções objetivo** (f): Consiste de um conjunto de funções de avaliação, onde cada função está associada a um objetivo do problema. Através de f obtém-se uma avaliação de uma solução $x \in \Phi$, sendo esta avaliação um ponto no espaço objetivo Ψ . Para o problema da mochila, $f = \{f_1(x), f_2(x), \dots, f_M(x)\}$, onde M representa o número de objetivos.
- **Relação binária** : representa uma comparação que permita ordenar as soluções e determinar quando uma solução é melhor que outra. Considerando a existência de múltiplos objetivos, as relações $<$, \leq ou $>$, \geq não permitem comparar duas soluções. Em geral, utilizam-se as relações de dominância.

4.1.2 Relação de Dominância

A comparação entre duas soluções em um problema com apenas uma função-objetivo pode ser realizada através das relações $<$, \leq ou $>$, \geq - seja o problema de minimização ou maximização. Por outro lado, a consideração de mais de uma função de avaliação - caracterizando um cenário multiobjetivo - necessita de uma relação que permita ordenar corretamente as soluções. Para tanto, utiliza-se a relação de dominância para comparação entre as soluções.

Considerando dois vetores de decisão x e y e suas respectivas avaliações $\psi^1 = f(x)$ e $\psi^2 = f(y)$, tem-se as seguintes definições na Tabela 2.

Supondo que o problema tem 4 objetivos e estes devem ser minimizados, então dadas as soluções $\psi^1 = (10, 11, 12, 13)$ e $\psi^2 = (9, 11, 12, 10)$: $\psi^1 \prec \psi^2$ diz-se que ψ^1 domina ψ^2 . Por outro lado, $\psi^2 \prec \psi^1$ é falsa, pois ψ^2 não domina ψ^1 . Essas relações são suficientes para que as soluções sejam comparadas entre si.

Além disso, a relação de dominância descrita na Tabela 2 possui duas propriedades importantes: **não-reflexividade**, ou seja, um vetor-objetivo ψ^1 não domina a si mesmo; e **assimetria**, onde $\psi^1 \prec \psi^2$ não implica em $\psi^2 \prec \psi^1$.

Um solução $x \in \Phi$ é dita **Pareto ótimo** se, somente se, não existir outra solução $y \in \Phi$, tal que $\psi^1 = f(y) \prec \psi^2 = f(x)$.

Tabela 2 – Tabela adaptada de (KNOWLES; THIELE; ZITZLER, 2006) ilustrando as relações entre vetores objetivos.

Relação	Interpretação no Espaço Objetivo	
domina estritamente	$\psi^1 \prec \prec \psi^2$	ψ^1 é melhor que ψ^2 em todos os objetivos
domina	$\psi^1 \prec \psi^2$	ψ^1 não é pior que ψ^2 em todos os objetivos e melhor em pelo menos um
domina fracamente	$\psi^1 \preceq \psi^2$	ψ^1 é não pior que ψ^2 em todos os objetivos
incomparável	$\psi^1 \psi^2$	nem $\psi^1 \preceq \psi^2$, nem $\psi^2 \preceq \psi^1$
indiferente	$\psi^1 \sim \psi^2$	ψ^1 tem o mesmo valor que ψ^2 em cada objetivo

O conjunto de todas as soluções Pareto ótimo é denominado **conjunto pareto ótimo**. Os vetores-objetivo associados as soluções do conjunto de Pareto ótimo são denominados **fronteira de Pareto ótimo**. O ideal durante o processo de busca é encontrar tais soluções que formem a fronteira de Pareto ótima. Entretanto, nem sempre é possível alcançar tal meta em tempo de processamento factível.

Diante da complexidade de muitos problemas de otimização, é aceitável aplicar técnicas de otimização que encontram **conjuntos de aproximação** - formado por soluções não dominadas entre si. Embora os conjuntos de aproximação não contenham todas as soluções Pareto ótimo, eles possuem soluções de alta qualidade e são produzidos em tempo computacional aceitável. O mapeamento dos vetores de decisão do conjunto de aproximação para vetores-objetivo define a **fronteira de aproximação**.

4.1.3 Conjuntos de Aproximação

Há inúmeras abordagens aplicadas à busca de conjuntos de aproximação. Estas abordagens, em geral, visam encontrar conjuntos de aproximação em tempo de processamento aceitável. Além disso, buscam caracterizar os conjuntos com soluções diversificadas e que estejam o mais próximo possível da fronteira ótima.

É crucial que a abordagem aplicada seja capaz de encontrar um conjunto de soluções que seja bem distribuído ao longo da fronteira ótima - mesmo que as soluções não sejam parte da fronteira ótima.

A busca pela diversidade na construção do conjunto de aproximação justifica-se pelo fato de os otimizadores considerarem, em geral, todos os objetivos do problema com igual importância. Assim sendo, um conjunto bem distribuído ao longo da fronteira de Pareto traz uma representação melhor de todo o espaço de busca. Em (KNOWLES, 2002), discute-se meios para manter a diversidade durante o processo de busca: *crowding*, *crowding* determinístico, pré-seleção e compartilhamento de *fitness*.

Outra característica marcante nos otimizadores presente na construção dos conjuntos de aproximação é o **elitismo**. A ideia é manter as melhores soluções encontradas du-

rante a busca, utilizando-as para reprodução de novas soluções. Sua utilização consiste basicamente em aplicar as soluções de elite para criar novos indivíduos na expectativa de que estes novos indivíduos sejam também incluídos no conjunto de soluções de elite (KNOWLES, 2002).

4.1.4 Conjuntos de Aproximação e Relação de dominância

As relações de dominância ilustradas na Tabela 2 são estendidas para avaliação de conjuntos de aproximação. Deste modo, é possível verificar se um conjunto de aproximação A domina um conjunto de aproximação B . Na Tabela 3, as relações de dominância aplicadas a conjuntos de aproximação são ilustradas.

Tabela 3 – Extensão das relações de dominância aplicadas a conjuntos de aproximação.

Relação		Interpretação no Espaço Objetivo
domina estritamente	$A \prec\prec B$	todo $\psi^2 \in B$ é estritamente dominado por pelo menos um $\psi^1 \in A$
domina	$A \prec B$	todo $\psi^2 \in B$ é dominado por pelo menos um $\psi^1 \in A$
melhor	$A \triangleleft B$	todo $\psi^2 \in B$ é fracamente dominado por pelo menos um $\psi^1 \in A$ e $A \sim B$
domina fracamente	$A \preceq B$	todo $\psi^2 \in B$ é fracamente dominado por pelo menos um $\psi^1 \in A$
incomparável	$A \parallel B$	nem $A \preceq B$ ou $B \preceq A$
indiferente	$A \sim B$	$A \preceq B$ e $B \preceq A$

Tais relações podem ser utilizadas para avaliação de otimizadores ao nível de conjuntos de aproximação gerados. Além disso, o *rank* de dominância utilizado para avaliação de conjuntos de aproximação emprega relações de dominância em nível de conjuntos - mais detalhes na seção 4.2.1.

4.2 Avaliação de Conjuntos de Aproximação

Os otimizadores aplicados à busca de conjuntos de aproximação em geral são estocásticos. Assim sendo, não produzem sempre o mesmo conjunto ou não há garantias de que será encontrada a fronteira de Pareto ótima. Portanto, deve-se avaliar a diversidade dos conjuntos gerados, o quanto as soluções geradas estão próximas do fronteira ótima e o esforço computacional para tal empreitada. Para solucionar tal problema, os **indicadores de qualidade** e **rank de dominância** são aplicados.

4.2.1 Rank de Dominância

O *rank* de dominância é uma metodologia para avaliação de otimizadores através da atribuição de um valor de *rank* para as amostras coletadas dos algoritmos. Neste cenário, um conjunto de aproximação gerado por um otimizador é considerado uma amostra. Assim sendo, cada conjunto de aproximação será associado a um valor de *rank* dentre todos os outros conjuntos de aproximação.

Considere O otimizadores que geram conjuntos de aproximação de forma estocástica. Cada um destes otimizadores é executado um número x de vezes. Assim sendo, cada otimizador gerou x conjuntos de aproximação A_i^j , $i = \{1, \dots, |O|\}$ e $j = \{1, \dots, x\}$.

De acordo com Knowles, Thiele e Zitzler (2006), é possível utilizar a relação melhor (\triangleleft) definida na Tabela 3 para fazer a atribuição do *rank* de dominância dos conjuntos gerados pelos otimizadores. Para tanto, considere-se um conjunto $\Lambda = \{A_1^1, \dots, A_i^j\}$. Desse modo o *rank* de um conjunto de aproximação A_i^j pode ser definido como:

$$\text{rank}(\Lambda_i) = 1 + |\Lambda_j \in \Lambda : \Lambda_j \triangleleft \Lambda_i|. \quad (4.3)$$

A Equação 4.3 permite que um valor de *rank* seja associado a cada conjunto de aproximação obtido por cada um dos otimizadores. Na definição dada, quanto menor o valor de *rank* melhor qualificado será o conjunto de aproximação. Deste modo, os conjuntos de aproximação de um dado otimizador serão transformados em uma amostra. Esta amostra consiste de x valores de *rank*.

A transformação aplicada através do rank de dominância permite que os otimizadores sejam avaliados pelos métodos estatísticos para checar se há diferença significativa entre eles.

4.2.2 Indicador de Qualidade

Um indicador de qualidade é um mecanismo prático e simples de atribuir um valor de qualidade a um conjunto de aproximação. Dado um conjunto de aproximação A e um indicador de qualidade I , pode-se definir um valor real que determina a qualidade de A .

A Equação 4.4 ilustra o mapeamento de um conjunto de aproximação para um valor real - caso unário. Deste modo, é possível atribuir valores que qualificam os conjuntos gerados por um otimizador multiobjetivo. Além disso, é possível aplicar testes estatísticos convencionais aos valores de qualidade dos conjuntos após aplicação dos indicadores.

$$I : A \longrightarrow \mathbb{R} \quad (4.4)$$

É possível definir um indicador de qualidade desde que se garanta algumas propriedades para o mesmo: monotonicidade; invariância de escala; esforço computacional; conhecimento do problema (ZITZLER; KNOWLES; THIELE, 2008).

A **monotonicidade** está relacionada a estrutura de preferência para comparação de soluções (relação de dominância, por exemplo). O indicador deve ser coerente com tal estrutura. Supondo que quanto maior o valor reportado pelo indicador melhor será o conjunto, então: $\forall A, B \quad A \prec B \Rightarrow I(A) > I(B)$.

A **invariância de escala** diz respeito as transformações aplicadas nos vetores-objetivo ($\Re \rightarrow \Re$). Estas transformações de escala não podem afetar o resultado obtido pelo indicador.

O indicador de qualidade deve ser obtido com baixo tempo de processamento e ser independente do problema no qual ele é utilizado.

4.2.2.1 Hipervolume

O **hipervolume** (H) é um indicador de qualidade proposto por Zitzler e Thiele (1999). A ideia por trás do indicador é verificar o tamanho do espaço objetivo que é dominado por um conjunto de aproximação.

O indicador aponta que um conjunto de aproximação A é melhor que um conjunto de aproximação B , quando o valor de hipervolume obtido para A é maior que o valor obtido para B - $I_H(A) > I_H(B)$.

A computação dos hipervolumes dos conjuntos de aproximação comparados tem como requerimento a definição de um ponto no espaço objetivo, que seja estritamente dominado por todos os vetores-objetivo dos conjuntos de aproximação envolvidos. Este ponto é denominado **ponto de nadir**.

Na Figura 10, há um exemplo de hipervolume para um caso onde há duas funções objetivo f_1 e f_2 cujos valores devem ser minimizados. O ponto de nadir é $p = (7, 7)$ e o conjunto de soluções que determinam o conjunto tem os seguintes vetores-objetivo: $A = \{(7, 1); (3, 2); (2, 5); (1, 7)\}$. A área cinza representa o espaço dominado pelas soluções do conjunto A .

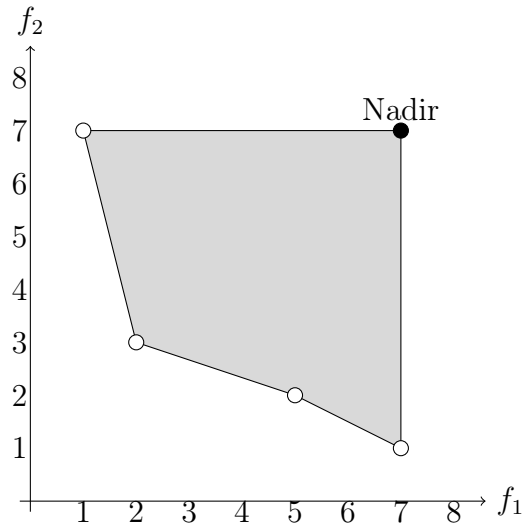
4.2.2.2 Epsilon

O indicador **Epsilon**, foi proposto por Zitzler et al. (2003). Os autores definiram duas versões do indicador: *epsilon* multiplicativo, I_ϵ , e *epsilon* aditivo, $I_{\epsilon+}$. Ambos são operadores binários.

Supondo um problema com o objetivos (todos para minimização), diz-se que um vetor-objetivo $\psi_1 \in \Psi$ ϵ -domina um vetor-objetivo $\psi_2 \in \Psi$, se:

$$\psi_i^1 \leq \epsilon \cdot \psi_i^2, \quad \forall 1 \leq i \leq o \quad \text{Caso multiplicativo} \quad (4.5)$$

Caso ψ^1 ϵ -domine ψ^2 , então deve existir um $\epsilon < 1$ de modo que a relação $\psi^1 \prec_\epsilon \psi^2$ seja verdadeira, assim como as condições da Equação 4.5.

Figura 10 – Exemplo gráfico de hipervolume de conjunto de aproximação.**Fonte:** própria.

Dado dois conjuntos de aproximação A e B , a ideia do indicador é utilizar um fator multiplicativo/aditivo que modifique os vetores de decisão do conjunto B de modo que ele ainda seja ϵ -dominado por A . Pode-se definir o indicador I_ϵ como (ZITZLER et al., 2003):

$$I_\epsilon(A, B) = \inf_{\epsilon \in \mathbb{R}} \{ \forall \psi^2 \in B \quad \exists \psi^1 \in A : \psi^1 \prec_\epsilon \psi^2 \} \quad (4.6)$$

O resultado obtido através do indicador I_ϵ mostra um fator pelo qual o conjunto pode ser considerado pior que outro. Por exemplo, na Figura 11 dois conjuntos de aproximação são ilustrados, $A = \{(2, 7), (5, 3), (7, 2), (3, 4)\}$ e $B = \{(3, 1), (2, 2), (1, 3)\}$:

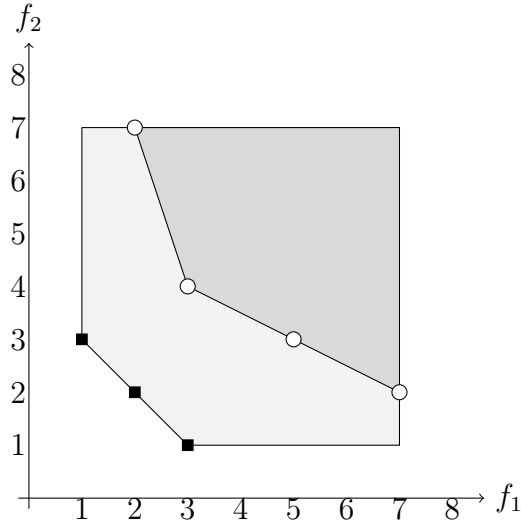
Considerando os conjuntos A e B e a minimização das funções-objetivo, o valor de $I_\epsilon(A, B) = 2,33$. Este fator indica o quanto o conjunto B precisa ser piorado para que ele seja ϵ -dominado pelo conjunto A . Por outro lado, $I_\epsilon(B, A) = 0,66$.

4.3 Algoritmos Evolucionários

Nesta seção, são apresentadas as principais técnicas de otimização multiobjetivo. Elas têm foco em evoluir uma população de indivíduos - algoritmos evolucionários.

4.3.1 Definições

Os algoritmos evolucionários são abstrações desenvolvidas visando mimetizar o processo evolucionário encontrado na natureza. Suas principais características são a criação e ma-

Figura 11 – Exemplos de conjuntos de aproximação para avaliação com indicador I_e .

Fonte: própria.

nutenção de uma **população** de indivíduos. Cada **indivíduo** da população é uma solução para o problema em questão. Além disso, dois processos importantes são incorporados: **crossover** (recombinação) e **mutação**.

O *crossover* é aplicado à criação de indivíduos e, em geral, envolve dois indivíduos pais. O resultado desta operação é um novo indivíduo (solução). Ele deve ser avaliado com respeito a sua inserção na população de indivíduos. Por outro lado, o processo de mutação ocorre em um determinado indivíduo. O objetivo é fazer uma pequena alteração no indivíduo de modo a promover diversidade na população.

Além disso, os algoritmos evolucionários também têm a adoção de critérios de **seleção de indivíduos**. Em cada iteração do algoritmo, é necessário criar novos indivíduos. Para tanto, indivíduos da população são selecionados para reprodução e a forma de selecionar os indivíduos também deve ser considerada na criação do algoritmo.

4.3.2 Abordagens Clássicas

As iniciativas pioneiras na otimização multiobjetivo não consideram o elitismo na construção das populações de soluções. Entretanto, trabalhos relevantes foram desenvolvidos e utilizados como base para evolução das técnicas de otimização multiobjetivo. Nesta seção, os algoritmos mais relevantes são apresentados.

O trabalho pioneiro de Schaffer (1984) deu origem ao *Vector Evaluated Genetic Algorithm* (VEGA). Esta abordagem considera um vetor que direciona a avaliação do *fitness* de um indivíduo. Deste modo, a busca é guiada considerando cada uma das funções do conjunto de funções do problema. Em cada iteração do processo evolucionário uma fração

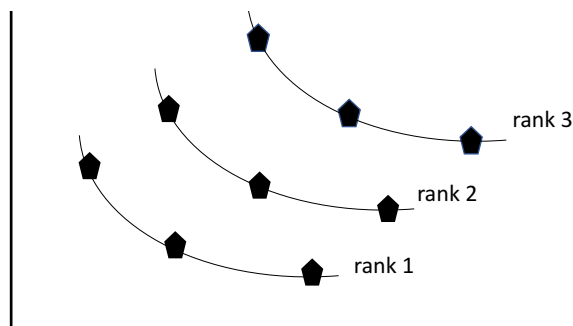
da população de indivíduos é otimizada com base em um dos objetivos do problema. O objetivo desta abordagem é otimizar a população em dimensões diferentes, mantendo os melhores indivíduos em cada uma das funções.

Fonseca e Fleming (1993) propuseram uma nova forma de avaliar a aptidão dos indivíduos de uma população. A ideia dos autores consiste em classificar os indivíduos da população com base na relação de dominância. O *fitness* do indivíduo é determinado pelo *rank* de dominância (ver Equação 4.3). O valor de *rank* é atualizado por iteração considerando todos os indivíduos da geração corrente. Os indivíduos não-dominados recebem valor 1 e são dominados por nenhum outro indivíduo da população.

Srinivas e Deb (1994) apresentaram um algoritmo genético baseado na classificação de indivíduos da população quanto as relações de dominância - *Non-Dominated Sorting Genetic Algorithm* (NSGA). O algoritmo proposto, segundo os autores, garante maior diversidade no conjunto de soluções não-dominadas, assim como maior aproximação da fronteira de Pareto ótima. Na Figura 12, há um exemplo de classificação dos indivíduos em fronteiras. Os indivíduos não-dominados por nenhum outro na população formam a primeira fronteira (*rank* 1). A segunda fronteira contém todos os não-dominados da população excluindo-se a fronteira de *rank* 1, e assim segue.

Em geral, O NSGA é diferente dos algoritmos descritos anteriormente no critério de seleção dos indivíduos. Além do ranqueamento das soluções em fronteiras de indivíduos não-dominados, conforme ilustrado na Figura 12, também se utiliza o compartilhamento de *fitness*. Um valor de *fitness* é associado a um indivíduo com base no seu *fitness* dividido pelo número de indivíduos próximos a ele. Portanto, a classificação dos indivíduos em fronteiras e o compartilhamento de *fitness* permitem que o algoritmo tenha rápida convergência e também diversidade na construção da fronteira de aproximação, respectivamente.

Figura 12 – Classificação da população em fronteiras de não-dominados.



Fonte: Própria.

O Algoritmo 1 ilustra o processo de classificação de soluções em uma população com P indivíduos.

Algoritmo 1 Algoritmo para classificação da população (DEB et al., 2002).

Entrada: População P **Saída:** População P'

```
1:  $S_p \leftarrow \emptyset$ 
2:  $n_p \leftarrow 0$ 
3: para  $p \in P$  faça
4:   para  $q \in P$  faça
5:     se  $p \prec q$  então
6:        $S_p \leftarrow S_p \cup \{q\}$ 
7:     senão se  $q \prec p$  então
8:        $n_p \leftarrow n_p + 1$ 
9:     fim se
10:   fim para
11:   se  $n_p == 0$  então
12:      $p_{rank} = 1$ 
13:      $F_1 \leftarrow F_1 \cup \{p\}$ 
14:   fim se
15: fim para
16:  $i \leftarrow 1$ 
17: enquanto  $F_i \neq \emptyset$  faça
18:    $Q \leftarrow \emptyset$ 
19:   para  $p \in F_i$  faça
20:     para  $q \in S_p$  faça
21:        $n_q \leftarrow n_q - 1$ 
22:     se  $n_q == 0$  então
23:        $q_{rank} \leftarrow i + 1$ 
24:        $Q \leftarrow Q \cup \{q\}$ 
25:     fim se
26:   fim para
27: fim para
28:    $i \leftarrow i + 1$ 
29:    $F_i \leftarrow Q$ 
30: fim enquanto
```

O primeiro laço, linhas 3 a 15, é utilizado para encontrar o conjunto de indivíduos não dominados por nenhum outro indivíduo presente na população P . Para cada solução p , um conjunto de indivíduos dominados é criado, S_p , caso $p \prec q$ ($q \in P$ e $q \neq p$) - linhas 5 e 6. Além disso, o número de indivíduos que dominam p é contabilizado e define seu *rank* na população - linha 8. Em caso de nenhum indivíduo da população dominar o indivíduo p , então p recebe *rank* igual a 1 e é adicionado ao conjunto de soluções não-dominadas, F_1 (linhas 12 e 13). O conjunto F_1 representa a fronteira de indivíduos não-dominados.

No laço das linhas 17 a 30, o Algoritmo 1 classifica as demais soluções em fronteiras de indivíduos não-dominados conforme ilustrado na Figura 12. Em cada iteração do laço mencionado, um conjunto de soluções Q é criado (linha 18). Este conjunto representa a i -ésima fronteira a ser classificada. A ideia é considerar a fronteira F_{i-1} inexistente.

Para isso, o seguinte mecanismo é aplicado: para cada indivíduo $p \in F_{i-1}$, considera-se os indivíduos dominados por p , que estão no conjunto S_p (linhas 19 a 27). Todos os indivíduos $q \in S_p$ terão o valor de *rank* n_q diminuído de 1 (linha 21). Logo, se o valor de n_q se tornar 0, então o indivíduo q será adicionado a fronteira i em construção (linhas 22 a 24). Por fim, o conjunto Q torna-se a fronteira F_i (linha 30).

4.3.3 Abordagens Elitistas

A busca por conjuntos de aproximação é uma tarefa árdua. Os algoritmos devem ser projetados considerando diferentes cenários. Dois aspectos importantes na busca são: diversidade e aproximação da fronteira de Pareto. Os algoritmos não-elitistas, apesar da importância, podem perder qualidade do conjunto de soluções a medida que a população evolui.

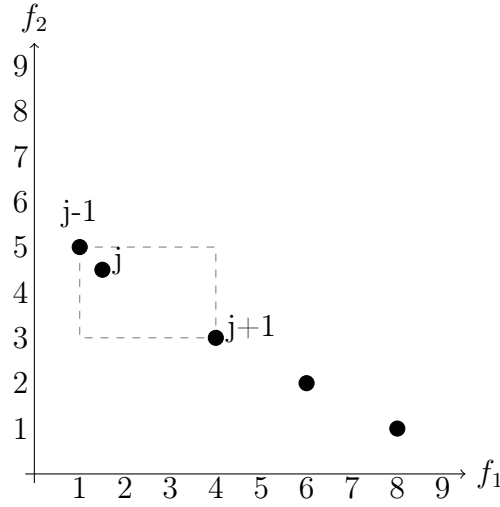
Por exemplo, considerando um algoritmo não-elitista com população de tamanho N e os processos de recombinação e mutação, é possível que um número maior que N de indivíduos não-dominados seja criado. Portanto, é necessário prever mecanismos de seleção que permitam escolher soluções que diversifiquem a população e aproximem da fronteira de Pareto. Várias abordagens foram propostas visando a tais objetivos.

4.3.3.1 *Non-Dominated Sorting Algorithm II* - NSGAII

Deb et al. (2002) propuseram uma versão elitista do NSGA. O NSGA2 realiza a mesma classificação dos indivíduos em fronteiras de indivíduos não-dominados, tal como o NSGA. Além disso, o procedimento emprega o conceito de *crowding distance* (distância de *crowding*). Este recurso é aplicado na seleção de indivíduos em duas situações: se todos os indivíduos são não-dominados, então se deve decidir que indivíduos devem entrar na população da geração seguinte; se a população da próxima geração está sendo criada e a última fronteira tem mais indivíduos do que a nova população ainda suporta, então alguns indivíduos serão eliminados.

A distância de *crowding* é calculada após a classificação da população em fronteiras de indivíduos não-dominados. Considerando a Figura 12, a distância de *crowding* é calculada utilizando os indivíduos da i -ésima fronteira. Para cada indivíduo j da fronteira F_i , o valor de *crowding* é calculado na geração corrente. Este valor é denominado $j_{distance}$. Graficamente, a $j_{distance}$ é um cubóide calculado ao redor de um indivíduo j com base nos seus vizinhos $j - 1$ e $j + 1$ conforme ilustrado na Figura 13.

O valor $j_{distance}$ e a classificação da população em fronteiras de não-dominados permitem a seleção de indivíduos para compor a população de cada geração de modo elitista. Além disso, é possível obter diversidade ao longo da fronteira de Pareto - isso justifica o uso do valor $j_{distance}$ dos indivíduos. Pois, através deste valor é possível verificar se uma dada região da fronteira possui muitos ou poucos indivíduos. Quanto maior o valor $j_{distance}$,

Figura 13 – Exemplo de região que determina a distância de *crowding* do indivíduo j .**Fonte:** própria..

maior é a distancia dos indivíduos $j - 1$ e $j + 1$ do indivíduo j . Isso é um indicativo da presença de poucos indivíduos na região analisada.

Esse mecanismo utilizado no NSGA2 permite que as populações de indivíduos selecionadas durante a evolução converjam para a fronteira ótima sem concentrar soluções em regiões específicas da fronteira de aproximação.

No Algoritmo 2, ilustra-se o processo de atribuição dos valores de distância de *crowding* às soluções da i -ésima fronteira de indivíduos, que é passada como parâmetro.

Algoritmo 2 Algoritmo para cálculo da distância de *crowding*.

Entrada: F_i **Saída:** F'_i

```

1: tam  $\leftarrow$  tamanho ( $F_i$ )
2: para  $p \in F_i$  faça
3:    $p_{distance} \leftarrow 0$ 
4: fim para
5: para  $f_i \in f$  faça
6:    $F_i \leftarrow ordenar\_por(F_i, f_i)$ 
7:    $max \leftarrow max(F_i, f_i)$ 
8:    $min \leftarrow min(F_i, f_i)$ 
9:    $F_i[1], F_i[tam] \leftarrow \infty$ 
10:  para  $pos = 2; pos < tam - 1; pos++$  faça
11:     $v1 \leftarrow valor(F_i[pos + 1], f_i)$ 
12:     $v2 \leftarrow valor(F_i[pos - 1], f_i)$ 
13:     $F_i[pos]_{distance} = F_i[pos]_{distance} + (v1 - v2)/(max - min)$ 
14:  fim para
15: fim para

```

O cálculo da distância de *crowding* é realizado para todos os indivíduos da fronteira F_i . Inicialmente, o valor $p_{distance}$ é definido como 0 para todos os indivíduos $p \in F_i$, linha 3. O valor da distância de *crowding* é acumulado para cada objetivo do problema e é obtido em relação aos vizinhos de um indivíduo. O laço das linhas 5 a 15 realiza o cálculo desse valor. O primeiro passo é ordenar os indivíduos da fronteira F_i de acordo com o valor da função-objetivo f_i (objetivo i) na linha 6. Em seguida, os valores máximo e mínimo para f_i são obtidos, linhas 7 e 8. Os indivíduos extremos do conjunto F_i de acordo com a função f_i recebem distância de *crowding* igual a infinito - linha 9. Em seguida, os demais indivíduos terão seu valor de *crowding* calculado. Para cada indivíduo na posição pos no conjunto F_i , o valor de *crowding* é calculado com base nos valores de f_i dos indivíduos $pos - 1$ e $pos + 1$ obtidos nas linhas 11 e 12. Por fim, o valor de distância do indivíduo na posição pos , $F_i[pos]$, é atualizado na linha 13. Isso é realizado para todos os objetivos.

Ademais, um operador baseado na distância de *crowding* foi desenvolvido pelos autores. Trata-se do operador de comparação baseado na distância de *crowding* (\prec_n) definido conforme a relação a seguir:

$$j_1 \prec_n j_2 \text{ sse } j_{1rank} < j_{2rank} \vee ((j_{1rank} = j_{2rank}) \wedge (j_{1distance} > j_{2distance})) \quad (4.7)$$

O operador definido na Equação 4.7 permite escolher os indivíduos mais aptos da população. Isso se deve ao fato de ele utilizar duas métricas. Primeiro, os melhores classificados quanto ao *rank* de dominância são selecionados. Supondo que o número de indivíduos não-dominados seja maior que o número de indivíduos que a população suporta, então alguns indivíduos devem ser removidos. Neste cenário, o *rank* de dominância não permite escolher de forma ideal quem deve entrar na população (dado que todos estão com o mesmo valor de *rank*). Logo, o operador utiliza a distância de *crowding* e permite escolher indivíduos melhor distribuídos na fronteira de aproximação.

No Algoritmo 3, O NSGA2 é ilustrado. Ele aplica os operadores ilustrados nos Algoritmos 1 e 2. A estratégia de busca consiste nas seguintes fases: criação de uma população inicial, linha 2, e evolução da população, que compreende o laço das linhas 4 e 18. Durante a evolução, uma população R é criada considerando a população da geração i e os filhos gerados por ela, linha 5. O população R é classificada em fronteiras de indivíduos não-dominados, *fronts*, utilizando Algoritmo 1 na linha 6. A população da geração seguinte, $i + 1$, é definida como vazia na linha 7. A partir disso, a população $i + 1$ será definida através da adição das fronteiras F_i , no laço das linhas 9 a 13 - observe-se que o número de membros deve ser no máximo N . Caso a última fronteira tenha mais indivíduos que ainda é suportado, $|pop_{(i+1)}| + |F_j| > N$, então o laço é encerrado e os membros da j -ésima fronteira são ordenados utilizando o operador definido na Equação 4.7, linha 14. Os $N - |pop_{(i+1)}|$ primeiros indivíduos serão adicionados à população $pop_{(i+1)}$ completando o número total de indivíduos, linha 15. Uma nova geração de filhos é criada e todo o processo se repete até que a condição de parada seja atingida, $i \geq max_iteracoes$.

Algoritmo 3 Non-Dominating Sorting Genetic Algorithm - NSGA2.

Entrada: Número de indivíduos da população N **Saída:** População de indivíduos P

```
1:  $i \leftarrow 1$ 
2:  $pop_i \leftarrow \text{inicializacao\_populacao}()$ 
3:  $offspring_i \leftarrow \text{recombinar}(pop_i)$ 
4: enquanto  $i < \text{max\_iteracoes}$  faça
5:    $R \leftarrow pop_i \cup offspring_i$ 
6:    $fronts \leftarrow \text{classificar\_populacao}(R)$ 
7:    $pop_{(i+1)} \leftarrow \emptyset$ 
8:    $j = 1$ 
9:   enquanto  $|pop_{(i+1)}| + |F_j| \leq N$  e  $F_i \in fronts$  faça
10:     calcular_distancia_crowding( $F_i$ )
11:      $pop_{(i+1)} \leftarrow pop_{(i+1)} \cup F_i$ 
12:      $j \leftarrow j + 1$ 
13:   fim enquanto
14:   ordenar( $F_j, \prec_n$ )
15:    $pop_{(i+1)} \leftarrow pop_{(i+1)} \cup F_i[1 : (N - |pop_{(i+1)}|)]$ 
16:    $offspring_{i+1} \leftarrow \text{recombinar}(pop_{(i+1)})$ 
17:    $i \leftarrow i + 1$ 
18: fim enquanto
```

4.3.3.2 S-Metric Selection Evolutionary Multiobjective Algorithm - SMS-EMOA

A abordagem SMS-EMOA foi proposta em (EMMERICH; BEUME; NAUJOKS, 2005) e (BEUME; NAUJOKS; EMMERICH, 2007). A ideia central do procedimento é utilizar o hipervolume como critério de seleção de indivíduos para remover da população os indivíduos menos aptos de acordo com as noções de elitismo e distribuição de solução ao longo da fronteira de aproximação.

O algoritmo é uma abordagem estacionária (*steady-state*), onde um novo indivíduo é gerado em cada iteração usando algum procedimento específico do problema. Este novo indivíduo gerado é incluído na população. A nova população é classificada em fronteiras de indivíduos de acordo com o Algoritmo 1 proposto por Deb et al. (2002). Em seguida, o hipervolume é utilizado como critério de seleção. O objetivo é eliminar o indivíduo que menos contribui para o crescimento do hipervolume.

Algoritmo 4 Algoritmo para otimização de Pareto com seleção baseada em hipervolume - SMS-EMOA.

```
1:  $pop \leftarrow \text{criar\_populacao}()$ 
2:  $i \leftarrow 0$ 
3: enquanto (não alcançar condição de parada) faça
4:    $q_{(i+1)} \leftarrow \text{novo\_individo}(pop_i)$ 
5:    $pop_{i+1} \leftarrow \text{reduce}(P_i \cup \{q_{i+1}\})$ 
6:    $i \leftarrow i + 1$ 
7: fim enquanto
```

No Algoritmo 4, o processo de otimização de Pareto utilizando o hipervolume como critério de seleção de indivíduos é descrito. O primeiro passo é criar uma população de indivíduos de tamanho N . Em seguida, essa população será evoluída por meio de algum critério de modificação de solução. Operadores de cruzamento e mutação são utilizados. Isso é realizado no laço das linhas 3 a 7. Para tanto, um novo indivíduo é criado na linha 4. Este indivíduo será adicionado a população corrente, pop_i . Neste momento, aplica-se o critério de redução da população através do hipervolume, linha 5. O procedimento **reduce** é o responsável pela classificação do conjunto de soluções em fronteiras e pela eliminação do indivíduo que menos contribui para o aumento do hipervolume.

Os autores apresentaram duas versões do algoritmo para reduzir a população (BEUME; NAUJOKS; EMMERICH, 2007). Os Algoritmos 5 e 6 são utilizados para redução de indivíduos. O primeiro algoritmo utiliza apenas a métrica do hipervolume para realizar a seleção. Por outro lado, o segundo algoritmo utiliza duas formas de seleção: hipervolume e outro critério (por exemplo número de indivíduos dominantes da solução avaliada). A principal razão por trás desta escolha é a complexidade de tempo do procedimento original. O algoritmo baseado apenas no hipervolume tem maior complexidade conforme se menciona no trabalho dos autores.

Algoritmo 5 Algoritmo de redução de indivíduos apenas com hipervolume.

Entrada: pop

Saída: pop'

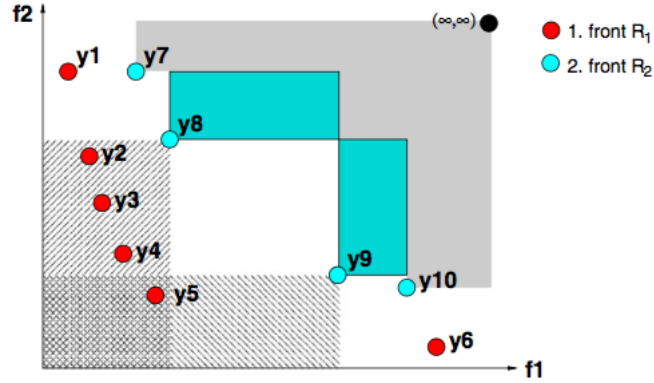
- 1: $fronts \leftarrow \text{fast_nondominated_sort}(pop)$
 - 2: $r \leftarrow \text{argmin}_{j \in F_i} [\Delta_H(j, F_i)]$
 - 3: $pop' \leftarrow pop \setminus r$
-

O Algoritmo 5 é utilizado para reduzir a população escolhendo um indivíduo da i -ésima fronteira após a classificação dos indivíduos. Uma população, pop , é dada como entrada e uma população, pop' , é retornada pelo procedimento. Os indivíduos são classificados em fronteiras de não-dominados, $fronts$, na linha 1. Em seguida, o indivíduo $j \in F_i$ que menos contribuir para aumentar o hipervolume da fronteira F_i será escolhido para ser eliminado, linha 2. O valor $\Delta_H(j, F_i)$ de um indivíduo j é obtido conforme a Equação 4.8 e deve ser minimizado. Portanto, para todo indivíduo $j \in F_i$, aquele que obtiver o menor valor $\Delta_H(j, F_i)$ é um candidato a ser removido da população, ou seja, quanto maior o valor $I_H(F_i \setminus j)$ maior a chance de j permanecer na população.

$$\Delta_H(j, F_i) = I_H(F_i) - I_H(F_i \setminus j) \quad (4.8)$$

Na Figura 14, considerando um problema com duas funções objetivo, f_1 e f_2 , é possível verificar o significado da Equação 4.8. Visualmente, apresenta-se o uso do critério de seleção baseado em hipervolume. Neste exemplo, duas soluções são candidatas a serem removidas, y_8 e y_9 . De acordo com a equação 4.8, a solução y_8 tem um valor menor que a

Figura 14 – Exemplo de uso do critério de seleção baseado em hipervolume e número de indivíduos dominantes.



Fonte: (BEUME; NAUJOKS; EMMERICH, 2007).

solução y_9 , ou seja, por ocupar uma área maior do hipervolume sua ausência terá maior impacto. Portanto, a solução y_9 será removida.

Esta versão do **reduce** ilustrada no Algoritmo 5 tem dois pontos importantes a serem observados. Primeiro, a complexidade do procedimento. Segundo, conforme pode ser observado na Figura 14, a solução y_9 é uma candidata a ser removida. No entanto, isso é o oposto do que se propõe com relação a diversidade na fronteira de aproximação. A solução y_9 está ocupando um local onde há poucos indivíduos e a mesma poderia, eventualmente, contribuir para novas soluções não-dominadas nesta região. Neste contexto, a segunda versão do procedimento **reduce** é empregada.

Algoritmo 6 Algoritmo de redução de indivíduos com dois critérios de seleção.

Entrada: pop

Saída: pop'

- 1: $fronts \leftarrow \text{fast_nondominated_sort}(pop)$
 - 2: **se** $|fronts| > 1$ **então**
 - 3: $r \leftarrow \text{argmin}_{j \in F_i}[d(j, pop)]$
 - 4: **senão**
 - 5: $r \leftarrow \text{argmin}_{j \in F_1}[\Delta_H(j, F_i)]$
 - 6: **fim se**
 - 7: $pop' \leftarrow pop \setminus r$
-

O Algoritmo 6 é utilizado para contrabalancear o uso do hipervolume como único critério de seleção de indivíduos. Ele se comporta de duas maneiras distintas. Primeiro, se todos os indivíduos são não-dominados, então o procedimento é exatamente igual ao Algoritmo 5. Por outro lado, se houver mais de uma fronteira de indivíduos, $|fronts| > 1$, então os indivíduos são analisados por algum critério diferente - os autores adotaram o

número de pontos dominantes, $d(j, pop)$. Essa versão do **reduce** classifica a população em fronteiras de indivíduos, $fronts$, linha 1. Considerando que $fronts > 1$, então o critério associado ao número de pontos dominantes é aplicado. O indivíduo j que possuir o maior valor de $d(j, pop)$ na fronteira $F_{|fronts|}$ será considerado para eliminação, linha 3. O valor $d(j, pop)$ de um indivíduo j é calculado conforme a Equação 4.9.

$$d(j, pop) = |\{y \in pop | y \prec j\}|. \quad (4.9)$$

O comportamento da segunda versão de **reduce** pode ser observado na Figura 14. Ao contrário do Algoritmo 5, que selecionou a solução y_8 , o Algoritmo 6 escolhe a solução y_9 . Esta escolha deve-se ao critério utilizado, número de pontos dominantes, definido na Equação 4.9. No exemplo ilustrado, o valor $d(y_8, F_2) = 4$, enquanto que o valor de $d(y_9, F_2) = 1$.

4.3.4 Abordagens não-elitistas

4.3.4.1 Generalized Differential Evolution - GDE3

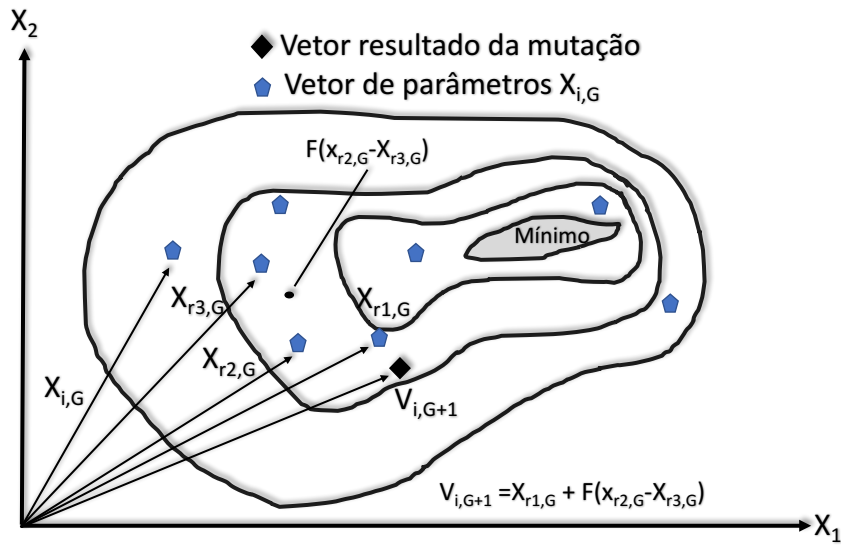
A abordagem *Generalized Differential Evolution* (GDE) pertence a uma família de algoritmos baseados em evolução diferencial. A proposta foi introduzida por Storn e Price (1997) denominada Differential Evolution (DE). Esta versão é aplicável a problemas de otimização com apenas uma função-objetivo e permite a otimização de espaços de busca contínuos.

O trabalho de Storn e Price (1997) introduziu uma nova abordagem para problemas de otimização. Considera-se uma população de N vetores de parâmetros em um espaço D -dimensional. Esta população de indivíduos passa por operadores de mutação e *crossover* para ser evoluída. Um indivíduo $x_{i,G}, i = 1...N$, é denominado vetor de parâmetros, sendo identificado pelo índice i . O índice G indica a geração a qual ele pertence. O comportamento do algoritmo é semelhante aos algoritmos evolucionários: uma população é inicializada; passa por um processo de evolução por meio de *crossover* e mutação e seleção de indivíduos de forma gulosa (melhor avaliado).

Na Figura 15, um exemplo de como o algoritmo utiliza a informação dos vetores de parâmetros é ilustrada em um espaço bi-dimensional. O exemplo trata da operação de mutação. Os índices associados a três indivíduos, $r1, r2$ e $r3$, são selecionados da população na geração G para que um novo indivíduo, $V_{i,G+1}$, seja definido. Observe-se que o novo indivíduo (vetor de parâmetros) é a combinação linear do vetores de parâmetros $X_{r1,G}, X_{r2,G}$ e $X_{r3,G}$. A composição de um novo indivíduo dá-se pelo Equação 4.10:

$$V_{i,G+1} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \quad (4.10)$$

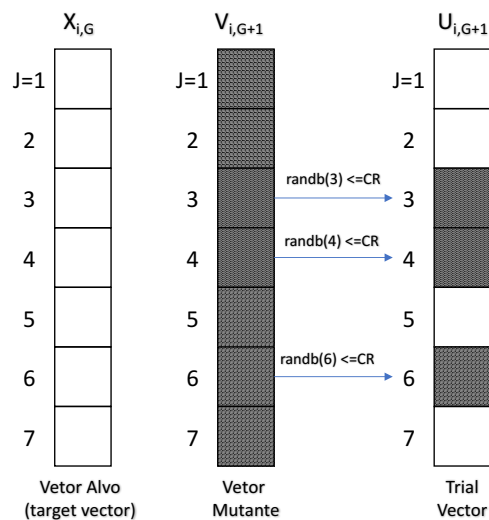
Figura 15 – Mutação utilizada no algoritmo DE.



Fonte: Adaptação de (STORN; PRICE, 1997).

A Equação 4.10 é definida como a mutação de indivíduos. O parâmetro F é ajustado durante a execução do algoritmo para controlar a velocidade e robustez da busca, segundo os autores.

Figura 16 – Crossover utilizado no algoritmo DE.



Fonte: Adaptação de (STORN; PRICE, 1997).

Na Figura 16, o processo de realização de *crossover* é ilustrado. Ele necessita das

informações do vetor de parâmetros original, $X_{i,G}$, e do vetor mutante, $V_{i,G+1}$, resultado da operação de mutação ilustrado na Figura 15 e Equação 4.10. Além disso, um parâmetro de *crossover*, $CR \in [0, 1]$, é necessário. O novo vetor de parâmetros é fruto da combinação de parâmetros dos vetores mencionados. Note-se que há um total de $D = 7$ parâmetros em cada vetor. A escolha dos parâmetros transmitidos para o novo indivíduo é determinada pela Equação 4.11:

$$U_{ji,G+1} = \begin{cases} V_{ji,G+1} & \text{if } (rand \leq CR) \vee j = j_{rand} \\ X_{ji,G} & \text{caso contrário} \end{cases} \quad \forall j = 1, \dots, D, j_{rand} \in \{1, \dots, D\} \quad (4.11)$$

A Equação 4.11 é utilizada para definir um novo vetor de parâmetros. O objetivo é misturar informações de ambos os indivíduos, $X_{ji,G}$ e $V_{ji,G+1}$. Entretanto, no pior caso possível quando $rand > CR$ para todo j , ainda é possível passar pelo menos uma informação de $V_{ji,G+1}$. Isso é assegurado pela condição $j = j_{rand}$ que garante pelo menos um parâmetro de $V_{ji,G+1}$ sendo passado adiante para $U_{ji,G+1}$.

O Algoritmo 7 resume o processo de otimização utilizando a combinação dos vetores de parâmetros como descrito anteriormente, linha 2. Uma população de vetores é gerada aleatoriamente considerando uma distribuição uniforme. Em seguida, o laço das linhas 3 a 11 é utilizado para controlar o número de gerações do processo evolucionário. Uma nova solução, $trial_{i,G+1}$, é gerada para cada vetor de parâmetro i por meio de mutação e *crossover*, linhas 5 a 7. $trial_{i,G+1}$ é avaliada contra o vetor $X_{i,G}$ e o vetor melhor avaliado é passado adiante na linha 8.

Algoritmo 7 Algoritmo DE original.

Entrada: x

Saída: y

```

1:  $G = 0$ 
2:  $pop_G \leftarrow \text{inicializa\_pop}(N)$ 
3: enquanto até atingir o critério de parada faça
4:   para  $i = 1, \dots, N$  faça
5:      $r1, r2, r3 \leftarrow \text{escolher\_indices}(pop_i)$ 
6:      $mutant_{i,G+1} \leftarrow \text{mutacao}(r1, r2, r3)$ 
7:      $trial_{i,G+1} \leftarrow \text{crossover}(mutant_{i,G+1})$ 
8:      $pop_{G+1} \leftarrow \text{melhor}(X_{i,G}, trial_{i,G+1})$ 
9:   fim para
10:   $G \leftarrow G + 1$ 
11: fim enquanto
```

O Algoritmo DE possui uma nomenclatura que resume algumas regras importantes de sua composição. Eles são denominados na literatura como: DE/ x / y / z . O valor de x representa o vetor de parâmetros que sofrerá o processo de mutação; o valor de y representa

o número de vetores utilizados no processo (na Figura 15 três indivíduos participam); z define o esquema de *crossover* (binário e real são valores comumente usados).

A adaptação da abordagem DE para resolução de problemas de otimização multiobjetivo foi realizada por diferentes autores como pode ser observado em (KUKKONEN; LAMPINEN, 2005). Neste trabalho, apenas O GDE é utilizado e, portanto, apenas este algoritmo e suas versões serão detalhadas. Em geral, as versões do GDE introduzem a capacidade de tratar mais de um objetivo, além de considerar problemas com restrições. Ademais, mecanismos de preservação de diversidade são incluídos no processo de otimização.

A primeira generalização do DE foi proposta por (KUKKONEN; LAMPINEN, 2004a) tendo como principal contribuição definir um critério de seleção de vetores dentro do contexto multiobjetivo. O critério de seleção é capaz de avaliar soluções que violam restrições do problema. Considerando o Algoritmo 7, tem-se que a escolha entre dois vetores, $trial_{i,G+1}$ e $X_{i,G}$, é realizada considerando apenas um valor. Entretanto, em uma avaliação multiobjetivo o cenário é diferente.

O vetor $trial_{i,G+1}$ é escolhido nos seguintes casos:

- se ele violar menos restrições que o vetor $X_{i,G}$;
- ou, $trial_{i,G+1}$ é viável e $X_{i,G}$ viola pelo menos um restrição;
- ou, simplesmente, $trial_{i,G+1}$ domina $X_{i,G}$;
- caso contrário, $X_{i,G}$ é escolhido.

A segunda generalização do DE está presente no trabalho de (KUKKONEN; LAMPINEN, 2004b). Neste trabalho, os autores tratam do cenário onde os vetores $trial_{i,G+1}$ e $X_{i,G}$ são mutuamente não-dominados. Logo, o critério de seleção de indivíduo adota a distância de *crowding* para decidir qual dos vetores passa à próxima geração. Os demais mecanismos da busca permanecem os mesmos que as versões GDE e DE. Basicamente, o que foi modificado é o terceiro item da lista descrita anteriormente. Em caso de não-dominância mutua entre $trial_{i,G+1}$ e $X_{i,G}$, então é verificado se a distância de *crowding* do vetor $X_{i,G}$ é menor, ou seja, está em uma parte do espaço de busca mais populado.

A terceira versão do GDE apresentada por (KUKKONEN; LAMPINEN, 2005) tem pequenas diferenças em relação as demais. No geral, apenas o critério de seleção foi modificado e também foi incluído o uso de classificação com base em não-dominância. O critério de seleção não considera a distância de *crowding* como na versão 2. Neste caso, quando $trial_{i,G+1}$ e $X_{i,G}$ são mutuamente não-dominantes, ambos os vetores passam para a próxima geração. Após o processo de criação de novos indivíduos por mutação e *crossover*, a população deve conter uma população maior que N . Logo, a população é reduzida a N vetores considerando o uso da classificação por não-dominância e a distância de *crowding* como no NSGA2.

Um ponto importante a ser discutido é a definição de vetores de parâmetros, que é a base da definição do Algoritmo DE e suas generalizações. Considerando um problema de otimização que possui D variáveis de decisão, então um vetor de parâmetros terá D parâmetros. Um parâmetro para cada variável de decisão. Dado que um problema de otimização tenha uma representação binária, a utilização do Algoritmo DE dá-se de forma direta (pode-se utilizar o padrão de projeto *Adapter* para tal manipulação (GAMMA et al., 1994)). Neste trabalho, a representação de indivíduos a nível de implementação não é realizada utilizando o formato binário. Por outro lado, utiliza-se conjuntos de árvores baseadas em caminho entre fontes e destinos, conforme será detalhado no capítulo 6.

4.3.4.2 *Multiobjective Evolutionary Algorithm based on Decomposition* - MOEA/D

Zhang e Li (2007) propuseram o *Multiobjective Evolutionary Algorithm Based on Decomposition* - MOEA/D. A ideia central da abordagem é dividir um problema multiobjetivo em N subproblemas de otimização agregando os valores-objetivo por meio de vetores de escalarização. Esta proposta, diferentemente das anteriores, não utiliza a classificação de indivíduos em fronteiras nem a distância de *crowding* para selecionar os indivíduos. Entretanto, utiliza-se a relação de dominância durante o processo de evolução de indivíduos da população.

Os autores consideram o fato de que uma solução ótima do Pareto pode ser uma solução ótima em um problema mono-objetivo considerando um vetor de escalares, sendo os valores das funções-objetivo agregados em um valor único. Partindo deste princípio, tem-se a decomposição do problema multiobjetivo. A aplicação de escalarização foi considerada em duas abordagens amplamente conhecidas: *Two-Phase Local Search* (TPLS) (PAQUETE; STÜTZLE, 2003) e *Multiobjective Local Search* (MOGLS) (JASZKIEWICZ, 2002).

A decomposição pode ser realizada por diferentes abordagens: soma ponderada, Tchebycheff entre outras (MIETTINEN, 1998). Neste trabalho, a escalarização adotada será o método de Tchebycheff. Considerando um vetor de escalares λ^1 e um indivíduo ind^1 , sua avaliação através do método de Tchebycheff é dado pela Equação 4.12, onde z^* contém os melhores valores para i -ésima função objetivo:

$$\text{Maximize } g^{te}(ind^1, \lambda^1) = \max_{1 \leq i \leq |f|} \{\lambda_i^1 | f_i(x) - z_i^* \} \quad (4.12)$$

Considerando a Equação 4.12, um problema multiobjetivo é considerado como N subproblemas no MOEA/D. Cada subproblema está associado a um vetor de escalares λ , e sua otimização é realizada considerando as informações de um conjunto de subproblemas que estejam próximos com base na distância euclidiana.

O Algoritmo 8 realiza a otimização de Pareto considerando os seguintes parâmetros de entrada: o número de subproblemas, N , o vetor de escalares, λ , e o número de vetores vizinhos, T (detalhes a seguir).

Algoritmo 8 Algoritmo evolucionário baseado em decomposição - MOEAD.

Entrada: N, λ, T

Saída:

```

1:  $EP \leftarrow \emptyset$ 
2: para  $\lambda^j \in \lambda$  faça
3:   para  $\lambda^l \in \lambda$  faça
4:      $DE^{jl} \leftarrow \text{dist\_euclidiana}(\lambda^j, \lambda^l)$ 
5:   fim para
6:   ordenar ( $DE^{jl}$ )
7:    $B(j) \leftarrow \text{obter\_indices}(DE^{jl}[1 : T])$ 
8: fim para
9: para  $\lambda^j \in \lambda$  faça
10:    $ind^j \leftarrow \text{criar\_individuo}(\lambda^j)$ 
11:    $FV^j \leftarrow \text{avaliar}(ind^j)$ 
12: fim para
13: para  $f_i \in f$  faça
14:    $z_i \leftarrow \min(pop, f_i)$ 
15: fim para
16: enquanto até atingir o critério de parada faça
17:   para  $i = 1$  até  $N$  faça
18:      $k, l \leftarrow \text{escolha\_dois\_individuos}(B(i))$ 
19:      $ind^* \leftarrow \text{recombinar}(k, l)$ 
20:     para  $f_i \in f$  faça
21:        $z_i \leftarrow \min(z_i, f_i(ind^*))$ 
22:     fim para
23:     para  $v \in B(i)$  faça
24:       se  $g^{te}(ind^*|\lambda^v, z) \leq g^{te}(ind^v|\lambda^v, z)$  então
25:          $ind^v \leftarrow ind^*$ 
26:          $FV^v \leftarrow \text{avaliar}(ind^*)$ 
27:       fim se
28:     fim para
29:      $EP \leftarrow \text{remover\_dominados}(EP, ind^v)$ 
30:      $EP \leftarrow EP \cup ind^*$ 
31:   fim para
32: fim enquanto

```

O MOEAD realiza uma série de procedimentos antes de iniciar o processo de otimização. Primeiro, a distância euclidiana é calculada para todos os pares de vetores, inclusive o cálculo é realizado de um vetor para ele mesmo. O laço das linhas 1 a 8 é utilizado para isso. Na linha 4, o valor da distância entre um vetor λ^j para o l -ésimo vetor é calculado e armazenado em DE^{jl} . Isso é realizado para todos os vetores $\lambda^j \in \lambda$. Na linha 6, os índices do vetores mais próximos de λ^j são ordenados e os T primeiros índices associados as menores distâncias para λ^j são armazenados no conjunto $B(i)$ na linha 7. O processo

de resolução do subproblema λ^j é realizado com informações do seus vizinhos, $B(i)$. Segundo, no loop das linhas 9 a 12, a população inicial é criada. Um indivíduo ind^j é criado para cada vetor λ^j , na linha 10. O processo de criação demanda algum operador específico do problema. O indivíduo criado é avaliado em cada uma das funções-objetivo do problema, cujos valores são armazenados em FV^j , linha 11. Por fim, um ponto ideal, z , é obtido a partir da avaliação de todos os indivíduos com respeito a i -ésima função-objetivo, na linha 14 (considere que todos os objetivos são de minimização).

A otimização dos N subproblemas é realizada até que uma condição de parada seja alcançada, linha 16. Em cada iteração do algoritmo, todos os N subproblemas são otimizados. Para cada subproblema i , os seguintes passos são realizados. Escolher dois índices, k e l , associados aos vizinhos de i , linha 18. Estes vizinhos serão utilizados para criar uma nova solução ind^* por meio de operadores de cruzamento e mutação (ou outro específico do problema), linha 19. Em seguida, o ponto ideal z é atualizado com os valores das funções-objetivo de ind^* , nas linhas 20 a 21. Neste momento, um processo importante do MOEAD é realizado. Trata-se da atualização dos vizinhos associados a solução do problema i , linhas 23 a 28. Para cada vizinho $v \in B(i)$, é verificado o valor obtido pelo método de Tchebycheff. Caso a nova solução obtenha um valor menor então a solução associada v é atualizada junto com seus valores objetivos linhas 25 e 26. Por fim, o arquivo de soluções externas EP é atualizado removendo-se as soluções dominadas por ind^* , que é adicionado a EP (caso não seja dominado por nenhum outro), linhas 29 e 30.

4.4 Algoritmos Transgenéticos

4.4.1 Fundamentos Biológicos

Os algoritmos Transgenéticos estão incluídos na classe de algoritmos evolucionários. Entretanto, eles divergem quanto a forma de evolução empregada. Esta abordagem leva em consideração a cooperação entre indivíduos diferentes como forma de evolução.

Ela explora teorias modernas relacionadas a evolução. Sabe-se que a teoria evolutiva mais conhecida e amplamente aceita é a de Darwin: a evolução baseia-se na competitividade dos seres onde os mais aptos e adaptados sobrevivem e reproduzem passando a frente sua carga genética. Um contraponto à Darwin é a teoria de Lamarck que sustenta a adaptação dos indivíduos ao meio e seu modo de vida como forma de adquirir aptidão. Tais aquisições são repassadas aos descendentes.

Teorias alternativas sugerem alguns pontos não justificados na teoria de Darwin. Como sugere Behe (1996), os indivíduos acumulam informações genéticas ‘inúteis’ durante o processo de evolução. Logo, esse acúmulo poderia levar a um ‘salto’ evolutivo. Ainda nessas correntes alternativas, Kimura (1968) sugere que variações no DNA são neutras para a seleção. Isso abriria possibilidade para saltos evolucionários por intermédio de compar-

tilhamento de informações entre indivíduos de espécies diferentes e fusão de indivíduos simbiossiontes.

Um teoria importantíssima foi proposta por Margulis e Fester (1991). Trata-se da **teoria da endossimbiose** (Serial Endosymbiotic Theory - SET). Ela indica que as relações entre dois ou mais indivíduos de espécies diferentes podem ser vantajosas para os mesmos. Neste contexto, o **indivíduo endossimbionte** é um ser que habita o interior de outro indivíduo. Tal proposta viabiliza a ideia de herança de conjuntos genômicos proposta por Lamarck. De mogo geral, a teoria indica que a competição nos moldes Darwiniano não é a única via de evolução de indivíduos, mas parte de um todo que inclui a **cooperação**.

A SET considera a **transferência horizontal de genes** como mecanismo principal para compartilhamento de informações genéticas. Os principais mecanismos considerados para transferência são **transformação** que é a absorção imediata de material genético, a **conjugação** que ocorre por meio de contato físico e a **transdução** que é intermediada por vírus.

4.4.2 Fundamentos Computacionais

A Transgenética mimetiza o processo de evolução baseado na endossimbiose de indivíduos. Para tanto, ela considera um contexto envolvendo um hospedeiro, endossimbiossiontes nas células do hospedeiro e vetores para controlar o fluxo de informações intercambiadas entre os participantes (GOLDBARG; GOLDBARG, 2002).

A proposta se sustenta em três premissas básicas:

- contextualização da endossimbiose de forma computacional por meio de uma célula hospedeira populada por endossimbiossiontes;
- formação de indivíduos híbridos por modificações no hospedeiro e endossimbiossiontes;
- mimetizar a troca de informações por meio de agentes vetores.

Na metáfora Transgenética um **cromossomo endossimbionte** representa uma solução. A otimização do cromossomo representa a otimização do problema de otimização em questão e a troca de informações é realizada por **vetores transgenéticos** (GOLDBARG; GOLDBARG, 2002).

A arquitetura dos algoritmos Transgenéticos apresenta três elementos importantes:

- população de endossimbiossiontes;
- população de vetores transgenéticos;
- informação genética do hospedeiro.

As informações genéticas presentes no hospedeiro podem ser do mais variado tipo. Ela pode ser formada por partes distintas de uma solução completa do problema e pode ser aplicada em diferentes contextos a depender do vetor e procedimentos utilizados. Qualquer informação útil ao problema, seja ela concreta (parte da solução) ou abstrata (regra, limite do problema etc.), pode ser utilizada para enriquecer a fonte de informação genética do hospedeiro. Vale ressaltar que tais informações são obtidas *a priori* e *a posteriori*. No primeiro caso, as informações são obtidas durante o processo de inicialização do algoritmo. Já o segundo caso, utiliza a retroalimentação do algoritmo para atualizar e manter informações úteis descobertas durante o processo de busca.

Estas informações serão manipuladas pelos vetores transgenéticos durante o processo evolutivo. Formalmente, o vetor transgenético¹ é definido como:

$$\Upsilon = \{I, \omega\}. \quad (4.13)$$

A Equação 4.13 ilustra formalmente um vetor transgenético. Ele carrega uma informação a ser utilizada na formulação, I , e os procedimentos a serem aplicados. Na Tabela 4, ilustram-se os principais procedimentos utilizados:

Tabela 4 – Procedimentos utilizados nos vetores transgenéticos (GOLDBARG; GOLDBARG, 2002).





Procedimento	Descrição
Ataque (p_1)	Critério que determina se um dado cromossomo é passível de manipulação por meio de um vetor transgenético.
Transcrição(p_2)	Determina como a informação transportada pelo vetor será transferida para o cromossomo.
Bloqueio/Desbloqueio(p_3)	Determina um período de tempo no qual uma manipulação torna-se inviolável (tempo, iterações).
Identificação (p_4)	Determina as posições que limitam a operação de um vetor.

Os procedimentos presentes na Tabela 4 caracterizam os vetores transgenéticos. Eles podem ser os seguintes: plasmídio, transposon, vírus e plasmídio recombinado. Na Tabela 5, suas características são apresentadas.

Os vetores transgenéticos ilustrados na Tabela 5 permitem a realização do **fluxo intracelular** da proposta do algoritmo transgenético. Este tipo de evolução não faz uso de reprodução sexual para gerar novos indivíduos. No entanto, os algoritmos transgenéticos

¹ Nos trabalhos da literatura utiliza-se $\lambda = \{I, \Phi\}$, porém estes símbolos já foram utilizados aqui para outra notação.

Tabela 5 – Vetores Transgenéticos (GOLDBARG; GOLDBARG, 2002).

Vetor	Metáfora Biológica	Método	Característica	Informação
Vírus		$\omega = \{p_1, p_2, p_3\}$	obtem informação a partir do DNA hospedeiro	Trecho de DNA
Plasmídio		$\omega = \{p_1, p_2\}$	obtem informação a partir do DNA hospedeiro	Trecho de DNA
Plasmídio Recombinado		$\omega = \{p_1, p_2\}$	formação de informação envolve mais de um cromossomo	Trecho de DNA
Transposon		$\omega = \{p_1, p_2, p_4\}$	Cria estrutura de vizinha	Trecho de DNA

não estão limitados apenas ao uso dos vetores transgenéticos. É possível utilizar operadores no **fluxo extracelular** como *crossover* e mutação. Este último pode ser dispensado tendo em vista que os vetores transgenéticos podem realizar operações que permitem salto evolutivo.

O processo de evolução guiado pela compartilhamento de informações na Transgenética tem como base o uso do que se denomina **regras transgenéticas**. São três tipos de regras:

- **Regra do Tipo 1:** define que informações serão transportadas pelos vetores transgenéticos;
- **Regra do Tipo 2:** orienta a constituição de operadores de manipulação (vetores);
- **Regra do Tipo 3:** desempenha papel de controle geral definindo, por exemplo, o número de cromossomos testados para manipulação, possibilidade de ataque por plasmídio ou transposon.

No Algoritmo 9, o processo de busca por soluções de um problema de otimização através da abordagem Transgenética é ilustrado:

Algoritmo 9 Algoritmo Transgenético Geral.

Entrada: Problema P **Saída:** Indivíduo Sol_{best}

```

1:  $pop \leftarrow \text{iniciar\_populacao}(P)$ 
2:  $Banco \leftarrow \text{iniciar\_banco}(P, pop)$ 
3: enquanto condição de parada não satisfeita faça
4:    $vetores \leftarrow \text{criar\_vetores\_trans}(Banco)$ 
5:    $Sub_{pop} \leftarrow \text{manipular\_cromossomo}(pop, vetores)$ 
6:    $pop \leftarrow \text{atualizar\_populacao}(pop, sub_{pop})$ 
7:    $Banco \leftarrow \text{atualizar\_banco}(pop)$ 
8: fim enquanto

```

O Algoritmo 9 inicia o processo de busca criando uma população de cromossomos endossimbiontes na linha 1. Em seguida, o banco de informações *a priori* é criado na linha 2. Neste momento, o laço das linhas 3-7 é iniciado. Os vetores transgenéticos são criados na linha 4, onde se determina o modo como vão atuar, que tipo de informação vai ser transportada por eles entre outras características. Uma subpopulação de indivíduos é criada na linha 5 através de manipulações dos indivíduos da população, pop , por meio dos vetores, $vetores$. A população é atualizada considerando pop e sub_{pop} na linha 6 e o banco de informações é atualizado (informações *a posteriori*) na linha 7.

Embora o Algoritmo 9 não mencione o número de objetivos que são possíveis de serem tratados, várias abordagens usando a Transgenética foram realizadas para problemas de otimização multiobjetivo. No trabalho de Rocha, Goldberg e Goldberg (2007), um algoritmo transgenético para o problema da árvore geradora bi-objetivo é proposto. O algoritmo possui dois tipos de plasmídeo e um transposon. Nenhum método elitista é utilizado para manutenção da população. Maia, Goldberg e Goldberg (2014) apresentam um estudo do problema da árvore geradora quadrática bi-objetiva através de algoritmo transgenético. No trabalho de Almeida et al. (2014), duas hibridizações do algoritmo transgenético com os *frameworks* NSGAI e MOEAD são propostas. Ao NSGAI foi incorporado os operadores de transferência horizontal de genes com as devidas adaptações. Já o MOEA/D foi adaptado para suportar o uso dos vetores transgenéticos. Neste caso, foi utilizado um novo tipo de repositório de informações associadas a cada subproblema.

4.5 Resumo

Neste capítulo, os principais conceitos sobre otimização multiobjetivo foram abordados. As noções básicas foram apresentadas. Em seguida, apresentou-se a avaliação de conjuntos de aproximação. Por fim, foram apresentados os principais algoritmos evolucionais e a Transgenética.

5 Formulação Matemática

O presente trabalho considera o projeto de algoritmos para otimização de três problemas de roteamento *multicast* diferentes. O primeiro problema foi proposto por Chen et al. (2013) e apresentado no Capítulo 3. Dois novos modelos são apresentados. O primeiro modelo considera a otimização da capacidade residual considerando o custo de configuração de uma solução como limite, sendo incorporado por meio de restrição - Seção 5.1. O segundo modelo considera três funções-objetivo: custo, capacidade residual e contagem de *hops*, Seção 5.2.

5.1 Capacidade Residual sob Limite de Orçamento

O modelo matemático proposto nesta seção foi considerado em um estudo preliminar sobre o problema de roteamento *multicast* multissessão (ANDRADE; GOLDBARG; GOLDBARG, 2015). A proposta consiste em fazer a otimização da capacidade residual considerando o custo da solução como um limite, através de um conjunto de restrições.

A definição do modelo utiliza a terminologia apresentada no Capítulo 3, Seção 3.1.2. Considere as seguintes variáveis de decisão incorporadas ao modelo:

- $x_{ij}^{kd} \in \{0, 1\}$ - variável de decisão binária. Indica se há fluxo proveniente da fonte $s \in S_k$ para um vértice $d \in D^k$ utilizando a aresta (i, j) ;
- $y_{ij}^k \in \{0, 1\}$ - variável de decisão binária utilizada para marcar as arestas que compõem a k -ésima árvore de distribuição;
- $Z_{ij} \in \mathbb{Z}$ - variável inteira que representa o valor de capacidade residual de uma aresta (i, j) ;
- $Z \in \mathbb{Z}$ - variável inteira que representa o valor objetivo a ser maximizado.

As Equações 5.1 a 5.10 definem o modelo proposto:

$$\text{Maximizar: } Z = \min_{(i,j) \in E} \{Z_{ij}\} \quad \forall (i,j) \in E \quad (5.1)$$

sujeito à:

$$\sum_{i \in V} x_{is_k}^{kd} - \sum_{i \in V} x_{s_k i}^{kd} = -1 \quad \forall k \in W, \forall d \in D^k \quad (5.2)$$

$$\sum_{(i,j) \in E} x_{ij}^{kd} - \sum_{(i,j) \in E} x_{ji}^{kd} = 0 \quad \forall k \in W \text{ e } \forall d \in D^k \quad (5.3)$$

$$\sum_{\substack{(i,j) \in E \\ j \neq r_k, d}} x_{ij}^{kd} - \sum_{\substack{(i,j) \in E \\ j \neq s_k, d}} x_{ji}^{kd} = 1 \quad \forall k \in W, \forall d \in D^k \quad (5.4)$$

$$x_{ij}^{kd} \leq y_{ij}^k \quad \forall k \in K, d \in D^k \text{ e } \forall (i,j) \in E \quad (5.5)$$

$$\sum_{d \in D^k} x_{ij}^{kd} - y_{ij}^k \leq 0 \quad \forall k \in W \quad (5.6)$$

$$b_{(ij)} \geq \sum_{k \in W} y_{ij}^k \cdot t^k \quad \forall (i,j) \in E \quad (5.7)$$

$$b_{(ij)} - \sum_{k \in W} y_{ij}^k \cdot t^k = Z_{ij} \quad \forall (i,j) \in E \quad (5.8)$$

$$\sum_{(i,j) \in E} y_{ij}^k \cdot c_{(ij)} \leq B, \quad \forall k \in W \quad (5.9)$$

$$x_{ij}^{kd} \in \{0, 1\}, y_{ij}^k \in \{0, 1\} \text{ e } k \in W, \forall d \in D^k \quad (5.10)$$

A formulação ilustrada nas Equações 5.1 a 5.10 consiste da adaptação do modelo de (CHOPRA; TSAI, 2001) para o problema da Árvore de Steiner. A formulação apresentada pelos autores tem como base o problema de distribuição de *commodities* (FOULDS, 1981). A nova abordagem adiciona suporte à configuração de $|W| \geq 1$ árvores de distribuição *multicast*. O conjunto de Restrições 5.2-5.4 é utilizado para garantir que o fluxo sai do vértice-fonte, s_k , e chega aos vértices de destino, D^k . As Restrições 5.2 indicam que há fluxo de saída do vértice-fonte. As Restrições 5.3 tratam da conservação de fluxo em vértices intermediários. As Restrições 5.4 garantem fluxo de entrada nos vértices do conjunto D^k . As restrições 5.5 são utilizadas para indicar que uma aresta (i, j) será utilizada na k -ésima árvore de distribuição. O conjunto de restrições 5.6 é utilizado para garantir que nenhum vértice-folha fique presente nas árvores construídas. As Restrições 5.7 e 5.8 são utilizadas para controlar a utilização das arestas quanto a capacidade das mesmas. Na Restrições 5.7, o valor de capacidade residual das arestas é sempre maior ou igual a carga de tráfego que será alocada. Já as Restrições 5.8 garantem que o valor de capacidade residual, Z_{ij} , de cada aresta é exatamente igual a $b_{(i,j)}$ (valor original) menos o valor alocado, $y_{ij}^k \cdot t^k$.

O objetivo é otimizar o valor de Z conforme a Equação 5.1. O valor de Z representa a capacidade residual mínima da rede e é obtido em função da capacidade residual das arestas (Ver Definição 1, Seção 3.4). O controle do valor de capacidade das arestas é

garantido pelo conjunto de Restrições 5.7 e 5.8. Essas restrições asseguram que o valor de Z_{ij} seja igual ao valor de capacidade residual das arestas $b_{(i,j)}$. Além disso, o modelo incorpora o controle de custo final de uma solução através do conjunto de Restrições 5.9. Um valor limite B é definido como custo máximo para construção da solução e representa um parâmetro do modelo.

A ideia por traz do modelo é possibilitar a otimização do valor de capacidade residual considerando diferentes valores de custo associados. O modelo visa a união de interesses de otimização presentes nos modelos descritos no Capítulo 3. Nenhum dos modelos analisados incorpora os dois objetivos no modelo proposto.

5.2 Roteamento *Multicast* Multissessão Multiobjetivo

O segundo modelo proposto é uma abordagem inovadora em relação ao problema de roteamento *multicast*. Nesta proposta, três funções-objetivo são definidas e contemplam interesses totalmente diferentes. Trata-se da otimização do valor de capacidade residual, considerando também o custo da solução e o comprimento dos caminhos nas árvores de distribuição (*hops*). Os três objetivos têm o mesmo grau de importância.

A terminologia definida na Seção 3.1.2 será utilizada como base para construção do modelo. Os conjuntos de variáveis de decisão $x_{ij}^{kd} \in \{0, 1\}$ e $y_{ij}^k \in \{0, 1\}$ são utilizados com a mesma semântica do modelo definido na seção anterior.

Além disso, considere também as seguintes variáveis:

- $h^{kd} \in \mathbb{Z}$ - variável de decisão utilizada para guardar o comprimento do caminho da fonte para um vértice cliente $d \in D^k$;
- $R \in \mathbb{Z}$ - variável que guarda o valor de capacidade residual e deve ser maximizada;
- $H \in \mathbb{Z}$ - variável que guarda o maior valor de h^{kd} . O valor deve ser minimizado.

As variáveis definidas acima serão utilizadas para definição do conjunto de funções-objetivo do problema. Seja $f = \{f_Z, f_{custo}, f_H\}$, onde:

$$f_Z = Z \tag{5.11}$$

A Equação 5.11 ilustra a função-objetivo associada a maximização da capacidade residual. O valor é obtido em função do conjunto de restrições que controla a capacidade residual das arestas (detalhes adiante).

$$f_{custo} = \sum_{(i,j) \in E} y_{ij} \cdot c_{(i,j)}, \quad \forall k \in W. \tag{5.12}$$

A Equação 5.12 é utilizada para obter o custo de configuração de uma solução. O valor deve ser minimizado. Ele é calculado em função do conjunto de variáveis y_{ij}^k , que indica as arestas utilizadas em cada uma das árvores de distribuição. O valor de custo, $c_{(i,j)}$, de cada aresta é considerado por árvore.

$$f_H = \max\{h^{kd}\}, \quad \forall k \in W \text{ e } \forall d \in D^k. \quad (5.13)$$

A Equação 5.13 é utilizada para obter o valor de maior comprimento de todos os caminhos entre fonte-destino nas árvores que compõem a solução.

Estas três funções compõem o modelo e permitem avaliar uma solução obtida pelos algoritmos empregados. Dado uma solução $x \in \Phi$, seu vetor objetivo é obtido da seguinte forma:

$$\begin{aligned} \psi &= f(x) \\ &= f(\max(f_Z(x)), \min(f_{custo}(x)), \min(f_H(x))) \end{aligned} \quad (5.14)$$

Considerando os conjuntos de variáveis e as funções-objetivo definidas, o modelo completo é composto pelas Equações 5.15-5.23:

Otimizar: $f(\max(f_Z(x)), \min(f_{custo}(x)), \min(f_H(x)))$.

sujeito à:

$$\sum_{s \in S_k} \sum_{(s,j) \in E} x_{sj}^{kd} = |D^k|, \quad \forall k \in W. \quad (5.15)$$

$$\sum_{\substack{(i,j) \in E \\ j \notin S_k \cup D^k}} x_{ij}^{kd} - \sum_{\substack{(j,i) \in E \\ j \notin S_k \cup D^k}} x_{ji}^{kd} = 0, \quad \forall k \in W, \forall s \in S_k \text{ e } \forall d \in D^k. \quad (5.16)$$

$$\sum_{(i,d) \in E} x_{id}^{kd} - \sum_{(d,i) \in E} x_{di}^{kd} = 1, \quad \forall k \in W \text{ e } \forall d \in D^k. \quad (5.17)$$

$$x_{ij}^{kd} \leq y_{ij}^k, \quad \forall k \in W, \forall d \in D^k \text{ e } \forall (i,j) \in E. \quad (5.18)$$

$$\sum_{\substack{(i,s) \in E \\ s \in S_k}} y_{is}^k = 0, \quad \forall k \in W. \quad (5.19)$$

$$b_{(i,j)} - \sum_{k \in W} t^k \cdot y_{ij}^k \geq Z \quad \forall (i,j) \in E. \quad (5.20)$$

$$Z \geq 0. \quad (5.21)$$

$$h^{kd} = \sum_{(i,j) \in E} x_{ij}^{kd}, \quad \forall k \in W \text{ e } \forall d \in D^k. \quad (5.22)$$

$$\begin{aligned} x_{ij}^{kd} \in \{0, 1\}, y_{ij}^k \in \{0, 1\}, h^{kd} \in \mathbb{Z}, Z \in \mathbb{Z}, H \in \mathbb{Z} \\ \forall k \in W, \forall d \in D^k, \forall (i,j) \in E. \end{aligned} \quad (5.23)$$

As Restrições 5.15-5.17 são utilizadas para controlar o fluxo que passa pelas arestas saindo dos vértices-fonte para o conjunto de vértices de destino. O conjunto de Restrições 5.15 garante que a quantidade de fluxo que sai do conjunto de vértices-fonte, S_k , é igual ao número de vértices de destino. O conjunto de Restrições 5.16 garante conservação do fluxo nos vértices intermediários. O conjunto de Restrições 5.17 garante a chegada de fluxo nos vértices de destino. As Restrições 5.18 são utilizadas para marcar as arestas que compõem as árvores de distribuição. Quando há fluxo passando por alguma variável x_{ij}^{kd} , a variável y_{ij}^k é ativada. O conjunto de Restrições 5.19 garante que não haja fluxo de entrada em nenhum dos vértices-fonte. O controle de capacidade residual das arestas é realizado através do conjunto de Restrições 5.20, garantindo que o valor de capacidade menos a carga de tráfego é sempre maior ou igual ao valor do objetivo de otimização, Z . O valor de Z é no mínimo 0 garantindo que nenhuma aresta tenha sua capacidade violada, conforme define a Restrição 5.21. O conjunto de Restrições 5.22 é utilizada para contabilizar o comprimento dos caminhos entre os pares de fonte-destino em cada árvore.

5.3 Análise dos Modelos

Nos modelos matemáticos propostos, considera-se importante a presença do custo de configuração durante o processo de otimização dos recursos da rede com vistas a viabilização da comunicação do grupos.

O primeiro modelo, otimização da capacidade residual, é uma abordagem preliminar do estudo presente neste trabalho. Ele permitiu analisar situações conflitantes na estrutura do problema. Considerando a otimização do valor de Z , foi possível observar conflitos entre o aumento deste valor e o custo de configuração necessário. Ao passo que limitar o valor de custo permite obter bons valores de Z . Resumidamente, ele serviu de base para construção da proposta de modelagem final, que trata da otimização de Z e do custo de configuração com a mesma importância.

A natureza conflitante entre custo e Z foi base para construção do modelo multiobjetivo. Claramente, conforme observado no modelo anterior e experimentos realizados (Capítulos 6 e 7), há um *trade-off* importante. Sendo assim, é possível ter diferentes configurações de soluções que podem ser usadas em diferentes contextos de acordo com a demanda. De modo geral, a otimização de um dos objetivos, Z ou Custo, tem como consequência a degradação do outro.

Além disso, o modelo também incorpora um objetivo importante com relação a qualidade de serviço. Nesse modelo, o comprimento do caminho da mensagem, denominado *hop*, também é considerado na otimização. Este objetivo está associado ao *delay* na transmissão da informação da fonte para um vértice-destino. Ele também considera uma correlação do custo com Z . A influência dessa correlação é especialmente visível quando o modelo considera caminhos mais longos para preservar a capacidade das arestas, conse-

quentemente, aumentando o valor de Z . Do mesmo modo, diminuir o comprimento dos caminhos pode levar a redução do custo de configuração. Ademais, a minimização do número de *hops* nos caminhos tende a equilibrar o *delay* para todos os pares de vértices fonte-destino, pois a redução do valor máximo de *hops* implica que os demais pares de fonte-destino utilizam caminhos dentro deste limite.

Resumidamente, o modelo possui as seguintes características:

- otimizar a capacidade residual diante de aplicações com alta demanda de banda;
- otimizar o custo de configuração da solução tornando factível sua implantação;
- otimizar o comprimento dos caminhos entre fonte-destino. Permitindo ter diferentes soluções com relação à tolerância de *delay* (*hops* como medida associada ao *delay*).

Os objetivos de otimização escolhidos permitem contemplar diferentes interesses de otimização que foram estudados separadamente na literatura. Os trabalhos de Lee e Cho (2004) e Chen et al. (2013) consideraram a otimização do valor de capacidade residual sem incorporar diretamente no modelo o custo da solução. Do mesmo modo, vários trabalhos abordaram a otimização do custo de configuração. Em alguns casos, houve a consideração da qualidade de serviço como em (KANG; PARK; PARK, 2009).

5.4 Resumo

Neste capítulo, foram apresentados os modelos matemáticos propostos para o problema de roteamento *multicast* multissessão. Dois modelos foram apresentados: o primeiro modelo é uma versão mono-objetiva que considera a otimização do valor de Z sujeito a um limite de custo. O segundo modelo considera a otimização de três funções-objetivo: capacidade residual, custo e contagem de *hops*.

6 Algoritmos Propostos

Neste capítulo, os algoritmos propostos para os modelos discutidos no Capítulo 5 são apresentados. A Seção 6.1 contém os algoritmos aplicados ao modelo que otimiza capacidade residual sujeito a limite de custo. A Seção 6.2 contém a heurística *MMVD* aplicada ao modelo proposto por Chen et al. (2013). Por fim, na Seção 6.3, os algoritmos aplicados ao modelo multiobjetivo são apresentados.

6.1 Otimização de Capacidade Residual Sob Limite de Custo

Nesta seção, os algoritmos aplicados ao modelo definido nas Equações 5.1-5.10 no Capítulo 5 são apresentados. Este modelo visa a otimização da capacidade residual sujeito a limite de custo. Quatro algoritmos foram projetados: GRASP (*Greedy Randomized Adaptive Search Procedure*), ACO (*Ant System Optimization*), Algoritmo Genético e *Iterated Local Search* (ILS). Além disso, operadores de uso geral aplicados como busca local em árvores são também apresentados.

6.1.1 Operadores de Busca Local

Dois algoritmos de busca local foram desenvolvidos. O primeiro algoritmo faz a substituição sistemática de arestas de árvores que compõem a solução do problema. Este algoritmo pode otimizar tanto o valor de capacidade residual quanto reduzir o custo de uma solução. O segundo algoritmo permite otimizar o custo de configuração da solução por meio da criação-remoção sistemática de ciclos.

Algoritmo 10 Algoritmo de melhoria por Substituição de Arestas.

Entrada: T, G, W **Saída:** T'

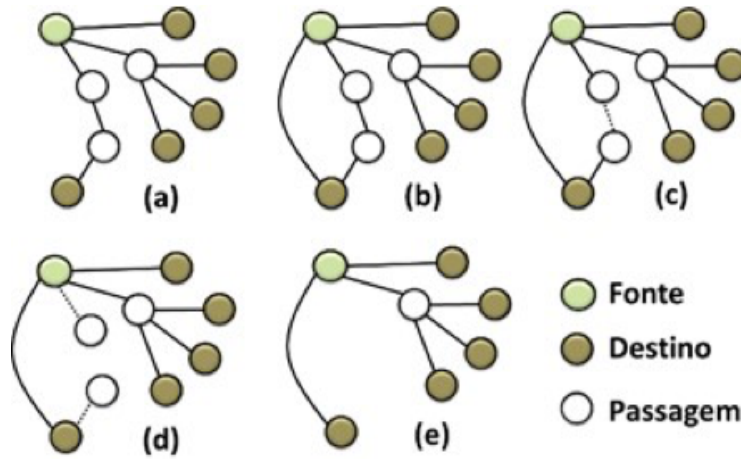
```
1:  $E_T \leftarrow \text{arestas}(T)$ 
2: para  $(i, j) \in E_T$  faça
3:   para  $T_k \in T$  faça
4:     se  $(i, j) \in T_k$  então
5:        $L \leftarrow \text{arestas\_disponíveis}(T_k, (i, j))$ 
6:        $(i, j)' \leftarrow \text{aresta\_aleatorio}(L)$ 
7:       substitui  $(T_k, (i, j)')$ 
8:     fim se
9:   fim para
10: fim para
```

O Algoritmo 10 inicia seu processamento obtendo o conjunto de arestas presentes em uma solução $T = \{T_1, T_2, \dots, T_{|W|}\}$ na linha 1. Em seguida, o laço das linhas 2-10 é

utilizado para procurar arestas que podem aumentar o valor de Z ou diminuir o custo da solução. Para cada aresta $(i, j) \in E_T$, verifica-se se a aresta está presente na árvore T_k (linha 4). Em caso positivo, um conjunto de arestas, L , que podem substituir (i, j) são obtidas na linha 5. Um aresta $(i, j)'$ é escolhida aleatoriamente na linha 6 e a mesma substitui (i, j) na árvore T_k na linha 7. Observe-se que a função de escolha de arestas, **arestas_disponíveis(.)**, determina que tipo de otimização será realizada: capacidade residual ou custo. No caso da otimização do custo, o conjunto de arestas L será formado por arestas que reduzem o custo mas não degradam o valor de Z .

O algoritmo de criação-remoção de ciclos é ilustrado através da Figura 17:

Figura 17 – Processo de substituição de arestas por criação-remoção de ciclos.



Fonte: própria.

Na Figura 17, o processo de criação-remoção de ciclos é ilustrado. Na Figura 17 (a) tem-se uma árvore para o k -ésimo grupo *multicast*. Em seguida, uma aresta é adicionada conforme ilustrado na Figura 17 (b). Esta aresta cria um ciclo que deve ser removido. Na Figura 17 (c), uma aresta é escolhida para remoção. Observe-se que esta aresta conecta dois vértices que não são clientes nem fonte do grupo. Após a remoção, a árvore fica como é ilustrada na Figura 17 (d). Esta remoção deixou dois vértices-folha que não são cliente nem fonte e, portanto, podem ser removidos. Na Figura 17 (e), ilustra-se o resultado final do processo.

O Algoritmo 11 recebe como entrada uma árvore T_k a ser otimizada, o conjunto de árvores da solução, T , o valor de consumo de banda, t^k , o conjunto de vértices-clientes, D^k , e o valor atual de capacidade residual, Z . Ele inicia o processamento criando dois conjuntos: E_{T_k} e V_k que representam as arestas e vértices da árvore T_k , respectivamente (linhas 1 e 2). Em seguida, o laço das linhas 3-11 é utilizado para criar e remover ciclos seguindo a proposta ilustrada na Figura 17. Um conjunto de arestas ligando vértices de V_k não presentes em E_{T_k} é criado e cada aresta é candidata a entrar na árvore T_k . Na

Algoritmo 11 Algoritmo de busca local baseado em ciclos.**Entrada:** T_k, T, t^k, D^k, Z **Saída:** T'_k

```

1:  $E_{T_k} \leftarrow \text{arestas}(T_k)$ 
2:  $V_k \leftarrow \text{vértices}(T_k)$ 
3: para  $(i, j) \in (V_k \times V_k) : (i, j) \notin E_{T_k}$  faça
4:   se  $(b_{(i,j)} - t^k) > Z$  então
5:      $C \leftarrow \text{cycle}(E_{T_k} \cup (i, j))$ 
6:      $(i, j)' \leftarrow \text{aresta\_para\_remover}(C)$   $\triangleright (i, j) : i, j \notin D^k \text{ e } i, j \neq s_K$ 
7:      $T'_k \leftarrow T_k \setminus (i, j)'$   $\triangleright \text{Remove } (i, j)' \text{ e vértice folha não terminal}$ 
8:      $Z \leftarrow \text{atualiza\_capacidade\_residual}(G, t^k)$ 
9:      $E_{T'_k} \leftarrow \text{edges}(T'_k)$ 
10:   fim se
11: fim para
12: se  $\text{tem\_melhoria}(T'_k)$  então
13:    $T_k \leftarrow T'_k$ 
14: fim se

```

linha 4, verifica-se a capacidade da aresta candidata (i, j) quanto a possibilidade de entrar em T_k . Se o seu valor de capacidade continuar maior que Z , então ela é considerada para criação do ciclo, C , na linha 5. Em seguida, uma aresta é removida do ciclo C na linha 6 conforme ilustrado na Figura 17(c). Após isso considere a operação realizada na linha 7 como os passos ilustrados na Figura 17(d) e (e). Nesse momento, além da redução do custo é possível que o valor de Z tenha melhorado, logo ele é atualizado na linha 8. O conjunto de arestas E_{T_k} é atualizado na linha 9. Por fim, se T'_k apresenta melhoria quanto a T_k , então há atualização na linha 13.

Um detalhe importante do processo de criação e remoção de ciclos é definido na linha 6 no comentário apresentado. A remoção de arestas é mais interessante quando $(i, j) : i, j \notin D^k$ e $i, j \neq s_K$. Pois, isso permite que mais de uma aresta seja removida da árvore possibilitando não somente a melhoria do custo, mas uma potencial melhoria do valor de Z .

6.1.2 Algoritmo baseado em colônia de formigas

Um algoritmo baseado em colônia de formigas foi projetado. A otimização por colônia de formigas tem sido amplamente utilizada em projeto de algoritmos para problemas mono e multi-objetivo. De modo geral, esta abordagem mimetiza o comportamento das formigas através de um modelo artificial de distribuição de feromônio durante o processo de busca. Tal abordagem permite que as formigas convirjam para uma solução de boa qualidade quando o modelo artificial de feromônio está bem ajustado (DORIGO; BLUM, 2005), (DORIGO; STÜTZLE, 2010).

A ideia proposta neste trabalho considera colocar uma formiga em cada vértice do

conjunto $D^k \cup S_k$ associado ao k -ésimo grupo *multicast* e permitir que as formigas se encontrem no grafo do problema formando uma árvore. Mecanismos para evitar ciclos são adicionados. Quando uma formiga se encontra com outra, apenas uma delas segue o percurso no grafo.

De modo geral, o algoritmo baseado em colônia de formigas funciona da seguinte forma:

- cada formiga inicia a busca com um conjunto de solução S_p vazio;
- em cada etapa a formiga adiciona um componente de solução ao conjunto solução S_p ;
- A escolha de um componente de solução é realizada com base na regra:

$$p(c_i^k | S_p) = \frac{\tau_{ij}^\alpha [\eta(c_i^j)^\beta]}{\sum_{c_i^j \in N(S_p)} \tau_{il}^\beta [\eta(c_i^l)^\beta]} \quad (6.1)$$

Os parâmetros α e β determinam que tipos de informações utilizadas (força):

- $\alpha = 0$: significa a escolha do próximo componente tem com base em informação heurística disponível sobre o problema;
- $\beta = 0$: significa que a escolha do próximo componente será estocástico guloso.

Além disso, o valor de feromônio que guia a busca por solução é atualizado conforme a Equação 6.2.

$$\eta_{ij} = (1 - \rho)\eta_{ij} + \sum_{s \in S_{upd} | c_i^j \in s} g(s) \quad (6.2)$$

Onde:

- S_{upd} : conjunto de soluções usadas para depósito de feromônio;
- ρ : parâmetro correspondente a taxa de evaporação (termos percentuais);
- $g(\cdot)$: função de avaliação da solução.

6.1.2.1 Construção de Solução

A construção de solução é ilustrada no Algoritmo 12. O algoritmo recebe como entrada o grafo do problema, G , o índice da árvore a ser criada, k , o conjunto de vértices clientes, D^k , o parâmetro h utilizado no procedimento **proximo_passo** (Algoritmo 13), e os parâmetros do modelo artificial de feromônio, α e β .

Algoritmo 12 Algoritmo de Construção de Árvore para o ACO.**Entrada:** $G, k, D^k, h, \alpha, \beta$ **Saída:** T_k

```

1:  $ants \leftarrow$  formigas ( $D^k \cup s_k$ )
2: enquanto  $ants.size() > 1$  faça
3:    $a \leftarrow$  formiga_aleatoria ( $ants$ )
4:    $pos\_atual \leftarrow$  posicao ( $a$ )
5:    $component \leftarrow$  proximo_passo ( $G, pos\_atual, h, \alpha, \beta$ )
6:   se validar_componente ( $component$ ) então
7:      $a.move()$ 
8:      $T_k \leftarrow T_k \cup component$ 
9:     se juntar_formigas ( $a, component$ ) então
10:        $ants.remove(a)$ 
11:   fim se
12: fim se
13: fim enquanto

```

O Algoritmo 12 inicia o processamento de criação de árvores definindo um conjunto de formigas artificiais, $ants$, na linha 1. Cada formiga inicia em um vértice $v \in D^k \cup S_k$. Em seguida, o laço das linhas 2-13 é utilizado para criar uma árvore para o k -ésimo grupo *multicast*. Na linha 3, uma formiga é escolhida aleatoriamente para realizar um movimento, que consiste em adicionar uma nova aresta à árvore. A posição atual, pos_atual , da formiga a é obtida na linha 4. O Algoritmo 13 é utilizado para fazer o movimento da formiga na linha 5. Caso o movimento seja válido, então a formiga o realiza na linha 7. O novo componente (aresta) é adicionado a árvore em construção na linha 8. Neste momento, um teste é realizado para verificar se mais de uma formiga chegou ao mesmo vértice, linha 9. Em caso afirmativo, remove-se a formiga a na linha 10. Quando todas as formigas se encontram, significa que o algoritmo construiu uma árvore.

6.1.2.2 Escolha de Componente de Solução

O Algoritmo 13 é utilizado para determinar a escolha do passo de uma formiga. Ele pode ser baseado na informação de feromônio já depositado pelas formigas, assim como pode ser uma escolha gulosa. Ele recebe como parâmetro o grafo do problema, G , h que determina a forma de escolha do passo e os parâmetros do modelo artificial de feromônio, α e β . O algoritmo devolve um componente e , que é uma aresta do grafo.

O processo de escolha de um novo componente realizado pelo Algoritmo 13 inicia com a determinação de um valor p no intervalo $(0.0, 1.0)$. Este valor é determinado de modo aleatório e uniforme. Caso $p < h$, então o componente será escolhido utilizando as informações de feromônio como em (DORIGO; STÜTZLE, 2010). Caso contrário, a escolha será feita de forma gulosa com base no custo do componente (aresta).

Algoritmo 13 Algoritmo de Escolha de Próximo Componente - **proximo_passo**.

Entrada: G, h, α, β **Saída:** e

- 1: $p \leftarrow \text{numero_aleatorio}(0, 1.0)$;
 - 2: **se** $p < h$ **então**
 - 3: $e \leftarrow \text{escolha_usando_feromonio}(G, \alpha, \beta)$
 - 4: **senão**
 - 5: $e \leftarrow \text{escolha_gulosa_custo}(G)$
 - 6: **fim se**
-

6.1.2.3 Algoritmo Proposto

No Algoritmo 14, ilustra-se o processo de otimização da capacidade residual sujeito a limite de custo. O algoritmo recebe como entrada os seguintes parâmetros: o grafo do problema, G , o conjunto de grupos *multicast*, W , o número máximo de iterações max_iter , os parâmetros do modelo artificial de feromônio, α e β , o parâmetro l que controla a atualização de feromônio após a construção de uma solução, o parâmetro θ que é utilizado para atualizar o valor de feromônio após a construção de uma árvore, o parâmetro h utilizado no Algoritmo 13 e o parâmetro γ que indica a probabilidade de aplicação dos Algoritmos 10 e 11.

Algoritmo 14 Algoritmo Baseado em Colônia de Formigas.

Entrada: $G, W, max_iter, \alpha, \beta, l, h, \theta, \gamma$ **Saída:** $T^{best} = \{T_1, T_2, \dots, T_{|W|}\}$

- 1: $T \leftarrow \text{solucao_inicial}(G, W)$
 - 2: $\text{atualiza_feromonio_global}(T, l)$
 - 3: $T^{best} \leftarrow T$
 - 4: **para** $i = 0$ **to** max_iter **faça**
 - 5: $T \leftarrow \emptyset$
 - 6: **para** $k \in W$ **faça**
 - 7: $T_k \leftarrow \text{constroi_arvore}(k, h, \alpha, \beta)$
 - 8: $T \leftarrow T \cup T_k$
 - 9: $\text{remove_arestas_congestionadas}(G, T_k)$
 - 10: $\text{atualiza_feromonio_local}(T_k, \theta)$
 - 11: **fim para**
 - 12: $prob \leftarrow \text{numero_aleatorio}(0, 1.0)$
 - 13: **se** $prob < \gamma$ **então**
 - 14: $\text{algoritmo_refinamento}(T, G, W)$
 - 15: **fim se**
 - 16: **se** $\text{melhorou_best}(T^{best}, T)$ **então**
 - 17: $\text{atualiza_feromonio_global}(T, l)$
 - 18: $T^{best} \leftarrow T$
 - 19: **fim se**
 - 20: **fim para**
-

O Algoritmo 14 inicia seu processamento criando uma solução inicial através do algo-

ritmo de Dijkstra (1959) na linha 1. Uma árvore enraizada na fonte é construída para cada grupo *multicast* $k \in W$. Em seguida, a tabela de feromônio é atualizada depositando a taxa l de feromônio em cada aresta da solução na linha 2. A solução T é definida como a melhor solução até então na linha 3.

O laço das linhas 4-20 é utilizado para realizar a otimização por colônia de formigas. Para cada grupo *multicast* $k \in W$, uma árvore é construída utilizando o Algoritmo 12. Esta árvore é incorporada a solução corrente, as arestas utilizadas são removidas do grafo para evitar congestionamento na construção da árvore seguinte (linhas 7, 8 e 9). O feromônio é atualizado localmente utilizando a taxa θ na linha 10.

Após a construção da solução T , é verificada a possibilidade de aplicação de um processo de refinamento nas linhas 12 a 14. Este processo utiliza os Algoritmos 10 e 11.

Em seguida, a solução T é comparada a solução T^{best} . É considerada melhoria se o valor de Z for maior que o valor atual ou se o valor de Z continuar o mesmo e o custo da solução tiver sido reduzido. Em qualquer um dos casos mencionados, o feromônio é atualizado na linha 17. Este processo repete-se até um número máximo, *max_iter*, de iterações.

6.1.3 Algoritmo baseado em GRASP

Nesta seção, um algoritmo baseado na abordagem GRASP (*Greedy Randomized Adaptive Search Procedure*) é ilustrado. O GRASP permite buscar por soluções de alta qualidade para problemas de otimização considerando um procedimento construtivo semi-guloso, que utiliza informações do problema para guiar a construção de solução. Além disso, o GRASP inclui um procedimento de busca local para intensificação da solução construída na fase anterior. Este processo de construção semi-guloso e busca local é repetido até que um critério de parada seja satisfeito (RESENDE; PARDALOS, 2006), (RESENDE; RIBEIRO, 2003).

6.1.3.1 Construção de Solução

Dois operadores de construção de soluções foram projetados. Os operadores são adaptações dos algoritmos de árvore geradora mínima e caminho mais curto. As adaptações visam a adicionar aleatoriedade ao processo de construção de árvores.

O Algoritmo 15 é baseado no algoritmo de árvore geradora de Kruskal. Ele recebe como parâmetros: o grafo do problema, G , o parâmetro μ utilizado para determinar a lista de arestas candidatas a compor a árvore em construção e a fonte associada ao grupo k , s_k .

O processo de construção de árvores do Algoritmo 15 inicia com a definição do conjunto L que representa as arestas do grafo G , linha 1. Em seguida, o laço das linhas 2-11 é utilizado para adicionar arestas à árvore em construção. Na linha 4, o tamanho da lista de

Algoritmo 15 Algoritmo para criação de árvores baseado em árvore geradora do GRASP adaptado de (MARTINS et al., 1999).

Entrada: G, μ, D^k, s_k

Saída: T_k

```
1:  $L \leftarrow \text{arestas}(G)$ 
2:  $\text{ordena\_por\_Z}(L)$  ▷ Ordem decrescente por valor de Z
3: enquanto  $L$  não é vazia faça
4:    $size \leftarrow \text{tamanho}(L) \cdot \mu$ 
5:    $pos \leftarrow \text{sorteia\_numero}(0, size)$ 
6:    $(i, j) \leftarrow L[pos]$ 
7:   se  $(i, j)$  está disponível então
8:      $T_k \cup (i, j)$ 
9:      $L \leftarrow L \setminus (i, j)$ 
10:  fim se
11: fim enquanto
12:  $\text{prune}(T_k)$ 
```

arestas candidatas relativo a μ é definido (parte das arestas de G são consideradas). Em seguida, uma aresta de L é escolhida dentre as primeiras $size$ arestas nas linhas 5 e 6. Em seguida, é verificada a disponibilidade da aresta na linha 7. Vale ressaltar que esta aresta pode estar removida para prevenção de sobrecarga. Caso seja possível utilizar a aresta, ela é incorporada a árvore T_k e removida de L nas linhas 8 e 9. No final do processo de construção, a árvore T_k é podada para remover vértices-folha que não pertencem a D^k ou s_k .

O Algoritmo 16 é utilizado para criar árvores associadas ao k -ésimo grupo *multicast*. A ideia central do procedimento é iniciar uma árvore com um caminho entre o vértice-fonte e um vértice-cliente. Em seguida, cada vértice-cliente não presente na árvore em construção será ligada a ela por um caminho. Este caminho é realizado entre um vértice $v \in T_k$ e um vértice $d \in D^k : d \notin T_k$. Os vértices são escolhidos aleatoriamente para garantir a geração de árvores diferentes. Os parâmetros que o algoritmo recebe como entrada são o grafo do problema, G , o vértice-fonte, s_k , e o conjunto de vértices clientes, D^k . O algoritmo devolve uma árvore T_k .

O processo de construção de árvore por caminho mais curto dá-se da seguinte forma. Um caminho P é criado considerando os vértices s_k e $d \in D^k$ escolhidos aleatoriamente, linha 1. Este caminho, arestas e vértices, é adicionado a árvore T_k na linha 2. O laço das linhas 3-8 é utilizado para adicionar os demais vértices de D^k à árvore em formação. Um vértice d é escolhido aleatoriamente em D^k na linha 4 e um vértice t presente na árvore também é escolhido aleatoriamente de T_k (este vértice pode ser um vértice comum.) na linha 5. Um novo caminho P é criado entre d e t na linha 6. O caminho é adicionado a árvore T_k na linha 7. Neste momento, evita-se que ciclos sejam criados. Este processo repete-se até que todos os membros de D^k estejam presentes na árvore. Por fim, a árvore T_k é podada para remover vértices e arestas não necessárias na linha 9.

Algoritmo 16 Algoritmo de Construção de Árvore baseado em Caminho mais Curto do GRASP.

Entrada: s_k, G, D^k

Saída: T_k

```

1:  $P \leftarrow \text{caminho\_mais\_curto}(s^k, \text{vertice\_aleatorio}(D^k))$ 
2:  $T_k \leftarrow T_k \cup P$ 
3: repeat
4:    $d \leftarrow \text{sorteia\_vertice}(D^k)$ 
5:    $t \leftarrow \text{sorteia\_vertice}(T_k)$ 
6:    $P \leftarrow \text{shortest\_path}(t, d)$ 
7:    $T_k \leftarrow T_k \cup P$ 
8: until Até  $D^k$  ser adicionado a árvore
9:  $\text{prune}(T_k)$ 

```

6.1.3.2 Algoritmo Proposto

No Algoritmo 17, o processo de otimização baseado no GRASP é ilustrado. Este procedimento utiliza os Algoritmos 15 e 16 para criação de solução e o Algoritmos 10 e 11 para realizar busca local. Ele recebe como parâmetros o grafo G , o conjunto de grupos *multicast*, W , o número máximo de iterações, max_iter , μ que é utilizado na construção de solução (Algoritmo 15) e um *heur* que define a probabilidade de utilização das heurísticas de construção de árvore.

Algoritmo 17 Algoritmo GRASP proposto.

Entrada: $G, W, \text{max_iter}, \mu, \text{heur}$

Saída: $T^{\text{best}} = \{T_1, T_2, \dots, T_{|W|}\}$

```

1: para  $i = 0$  até  $\text{max\_iter}$  faça
2:   para  $k \in W$  faça
3:      $\text{prob} \leftarrow \text{numero\_aleatorio}(0, 1.0)$ 
4:     se  $\text{prob} < \text{heur}$  então
5:        $T_k \leftarrow \text{arvore\_caminho\_mais\_curto}(G, D^k, s_k)$ 
6:     senão
7:        $T_k \leftarrow \text{arvore\_geradora\_restrita}(G, \mu, D^k, s_k);$ 
8:     fim se
9:      $T \leftarrow T \cup T_k$ 
10:  fim para
11:   $T' \leftarrow \text{busca\_local}(T, G, W)$ 
12:   $T^{\text{best}} \leftarrow \text{atualiza\_melhor}(T^{\text{best}}, T')$ 
13: fim para

```

A busca utilizando o Algoritmo 17 é realizada no laço das linhas 1-13. Uma solução é construída no laço das linhas 2-10. Para cada grupo *multicast* k , uma árvore é criada. As árvores vão constituir uma solução. Na linha 3, um valor, prob , é definido aleatoriamente. prob indica a possibilidade de usar uma das heurísticas de construção de árvores. Se $\text{prob} < \text{heur}$, o algoritmo baseado em caminho mais curto é utilizado na linha 5. Caso contrário, o algoritmo baseado em árvore geradora é utilizado na linha 7. Cada árvore

criada é adicionada a solução em construção na linha 9. A busca local é aplicada à solução T gerando uma nova solução T' na linha 11, que em seguida é verificada com relação a melhor solução encontrada na linha 12.

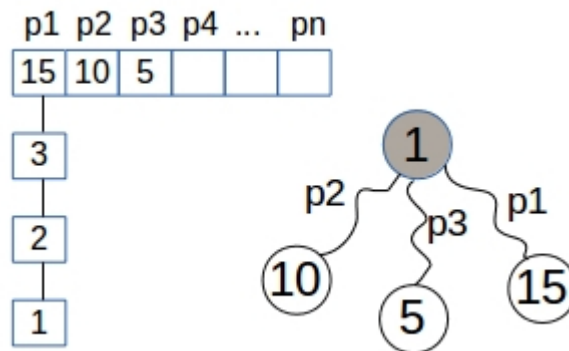
6.1.4 Algoritmo genético

Nesta seção, apresenta-se um algoritmo genético para a otimização da capacidade residual sujeito a um limite de custo. Os algoritmos genéticos mimetizam o processo evolucionário segundo os princípios defendidos por Darwin (GOLDBERG, 1989), (LUKE, 2009). No algoritmo proposto, a representação de solução utilizada é diferente daquelas usadas em algoritmos anteriores. Além disso, algoritmos de *crossover* e mutação são apresentados.

6.1.4.1 Representação de Solução

O *cromossomo* utilizado para representar uma solução no algoritmo genético é definido como um conjunto de caminhos entre cada par de vértices s_k e $d \in D^k$. Cada gene do cromossomo constitui um caminho entre a fonte um vértice de destino. Na Figura 18, ilustra-se uma árvore enraizada no vértice-fonte 1 com ligação a três vértices de destino 10, 5 e 15, cujos caminhos são $p1$, $p2$ e $p3$.

Figura 18 – Representação de uma solução: estrutura utilizada (cromossomo) e árvore associada a representação.



Fonte: própria.

Na Figura 18, um cromossomo é representado por uma lista de caminhos. Cada *slot* da lista é um caminho que liga o vértice-fonte a um vértice de destino. Cada árvore ocupa exatamente $|D^k|$ *slots* e são armazenadas na ordem de leitura das instâncias. Observe-se que esta ordem não é violada durante o processo de otimização.

6.1.4.2 Operador de Construção de Solução

A construção de solução é realizada através do uso do algoritmo de Dijkstra (1959). O Algoritmo 18 é utilizado para criação de árvores considerando a raiz da árvore escolhida de modo aleatório e uniforme. Ele recebe como parâmetros o grafo do problema, G , e o conjunto de grupos *multicast*, W . Uma solução é retornada contendo uma árvore para cada grupo.

Algoritmo 18 Algoritmo de Construção de Solução para Algoritmo Genético.

Entrada: G, W

Saída: $T = \{T_1, T_2, \dots, T_{|W|}\}$

```

1: cromossomo  $\leftarrow \emptyset$ 
2: para  $k \in W$  faça
3:    $root \leftarrow \text{vertice\_aleatorio}(s_k, D^k)$ 
4:    $T_k \leftarrow \text{arvore\_caminho\_mais\_curto}(root)$ 
5:   cromossomo  $\leftarrow \text{cromossomo} \cup T_k$ 
6: fim para
7:  $T \leftarrow \text{cromossoma}$ 

```

O Algoritmo 18 inicia a construção de solução definindo um cromossomo vazio na linha 1. Em seguida, o laço das linhas 2-6 é utilizado para construção das árvores que vão compor a solução. Um vértice raiz, $root$, é escolhido aleatoriamente no conjunto $s_k \cup D^k$ na linha 3. Uma árvore enraizada em $root$ é construída na linha 4 e adicionada ao cromossomo na linha 5. Observa-se que a árvore é adicionada conforme a representação ilustrada na Figura 18.

6.1.4.3 Operadores de Recombinação e Mutação

O Algoritmo 19 é utilizado para criação de um novo indivíduo por meio de cruzamento de dois indivíduos pais. Ele recebe como entrada duas soluções, T^1 e T^2 , e o conjunto de grupos *multicast*, W .

O cruzamento de soluções ilustrado no Algoritmo 19 inicia com a definição do tamanho do cromossomo na linha 1. Em seguida um cromossomo vazio é definido na linha 2 e um ponto no qual o cromossomo será dividido é definido na linha 3. Os laços das linhas 4-6 e 7-9 são utilizados para criar uma nova solução misturando os caminhos das soluções T^1 e T^2 . Os caminhos da solução T^1 são passados para a nova solução na linha 5 e os caminhos da solução T^2 são passados na linha 8. Por fim, o cromossomo é validado removendo ciclos e vértices não necessários na linha 10.

O mecanismo de mutação de solução é ilustrado no Algoritmo 20. A ideia central do algoritmo é remover arestas do grafo no qual a solução foi construída. Em seguida, os caminhos da solução que possuem as arestas removidas devem ser reformulados para retirar as arestas removidas do grafo.

Algoritmo 19 Algoritmo de Recombinação de solução

Entrada: T^1, T^2, W **Saída:** $T = \{T_1, T_2, \dots, T_{|W|}\}$

```

1:  $SIZE = \sum_{k \in W} |D^k|$ 
2:  $cromossoma \leftarrow \emptyset$ 
3:  $point \leftarrow \text{numero\_aleatorio}(SIZE)$ 
4: para  $i \leftarrow 0$  até  $point$  faça
5:    $cromossoma[i] \leftarrow T^1[i]$ 
6: fim para
7: para  $i \leftarrow point + 1$  até  $SIZE - 1$  faça
8:    $cromossoma[i] \leftarrow T^2[i]$ 
9: fim para
10:  $\text{validar}(cromossoma)$ 
11:  $T \leftarrow cromossoma$ 

```

O Algoritmo 20 é utilizado para realizar a mutação de indivíduos. Ele recebe como entrada uma solução, T , o grafo do problema, G , e ϱ que controla o tamanho da lista de arestas que serão removidas.

Algoritmo 20 Algoritmo de mutação do algoritmo genético.

Entrada: T, G, ϱ **Saída:** $T' = \{T_1, T_2, \dots, T_{|W|}\}$

```

1:  $SIZE = \sum_{k \in W} |D^k|$ 
2:  $L \leftarrow \text{arestas}(T)$  ▷ ordem crescente por capacidade residual
3:  $size \leftarrow |L|$ 
4:  $ER \leftarrow \emptyset$ 
5: para  $i \leftarrow size * \varrho$  até 0 faça
6:    $(i, j) \leftarrow \text{random}(L)$ 
7:    $G \leftarrow G \setminus (i, j)$ 
8:    $ER \leftarrow ER \cup (i, j)$ 
9: fim para
10:  $cromossoma \leftarrow \emptyset$ 
11: para  $i = 0$  até  $SIZE - 1$  faça
12:   se Existe aresta de  $ER$  em  $T[i]$  então
13:      $cromossoma[i] \leftarrow \text{shortest\_path}(T[i].source, T[i].dest)$ 
14:   senão
15:      $cromossoma[i] \leftarrow T[i]$ 
16:   fim se
17: fim para
18:  $T' \leftarrow cromossoma$ 

```

O processo de mutação do Algoritmo 20 inicia com a definição do tamanho do cromossomo na linha 1. Em seguida, uma lista de arestas da solução, L , é definida na linha 2. O conjunto de arestas a serem removidas é definido como vazio, ER , na linha 4. O laço das linhas 5-9 é utilizado para escolher arestas de L . Um cromossomo é definido na linha 10. Em seguida, o laço das linhas 11-17 é utilizado para reformular a solução.

Para cada caminho i no cromossomo, se houver alguma aresta de ER presente, então substitua o caminho (linhas 12 e 13). Caso contrário, apenas o caminho i é passado para o cromossomo alterado na linha 15.

6.1.4.4 Algoritmo Proposto

O Algoritmo 21 é utilizado para realizar a busca de soluções por meio de evolução de indivíduos através da abordagem baseada em algoritmo genético. Ele recebe como entrada o grafo do problema, G , o conjunto de grupos *multicast*, W , a taxa de *crossover*, $cross$, o número máximo de iterações, max_iter , a taxa de mutação, mut , um valor de probabilidade, l , que determina a aplicação dos operadores de busca local e o tamanho da população, pop .

O processo de evolução do Algoritmo 21 inicia com a criação da população inicial e definição da melhor solução nas linhas 1-3. Em seguida, o laço das linhas 4-9 é utilizado para evoluir a população, P . Para tanto, o operador de cruzamento é aplicado na linha 5 com uma taxa $cross$. Após isto, a mutação de indivíduos da população é realizada na linha 6 com uma taxa mut . Uma taxa l de indivíduos passa pelo processo de refinamento através de operadores de busca local (Algoritmos 10 e 11) na linha 7. Por fim, a melhor solução é atualizada na linha 8.

Algoritmo 21 Algoritmo Genético proposto.

Entrada: $G, W, cross, max_iter, mut, l, pop, \varrho$

Saída: $T = \{T_1, T_2, \dots, T_{|W|}\}$

```

1:  $P \leftarrow \emptyset$ 
2: inicializacao ( $P, pop$ )                                ▷ Algoritmo 18
3:  $T \leftarrow$  melhor_individuo ( $P$ )
4: para  $i \leftarrow 0$  ate  $max\_iter - 1$  faça
5:   recombinao( $P, cross$ )                                ▷ Algoritmo 19
6:   mutacao ( $P, mut, \varrho$ )                                ▷ Algoritmo 20
7:   refinamento ( $P, l$ )                                  ▷ Algoritmo 10
8:    $T \leftarrow$  melhor_individuo ( $P$ )
9: fim para
```

6.1.5 Algoritmo baseado em *Iterated Local Search*

A abordagem *Iterate Local Search* (ILS) permite buscar soluções através de mecanismos de perturbação de solução aliado a aplicação de busca local (LOURENÇO; MARTIN; STÜTZLE, 2003). Nesta seção, um algoritmo baseado em ILS é apresentado. O algoritmo possui um procedimento de criação de solução, que também pode ser utilizado para perturbação.

6.1.5.1 Construção de Solução

O Algoritmo 22 é utilizado em duas situações diferentes. Ele pode ser utilizado para construção de soluções, assim como pode ser utilizado para perturbar uma dada solução. Ele recebe como parâmetros o grafo do problema, G , o conjunto de grupos *multicast*, W , ϑ que é utilizado no procedimento de bloqueio de arestas, a lista de árvores que serão perturbadas na solução, BT , e a lista de arestas bloqueadas durante a construção de árvores ou busca local, BL , e o *iter* que indica quando aplicar o algoritmo para construir uma solução ou perturbá-la.

Algoritmo 22 Algoritmo para construção/pertubação de solução do ILS.

Entrada: $G, W, T, \vartheta, BL, BT, iter$

Saída: $T' = \{T_1, T_2, \dots, T_{|W|}\}$

```

1:  $T' \leftarrow \emptyset$ 
2: para  $k \in W$  faça
3:   se reutilizar(  $BT[k]$ ) então
4:      $T' \leftarrow T' \cup T_k$ 
5:   senão
6:     se  $iter = 0$  então
7:        $BL \leftarrow$  arestas_para_remove ( $T, \vartheta$ )
8:     fim se
9:     remove_aresta ( $G, BL, s_k, D^k, t^k$ )
10:    se  $iter \geq 1$  então
11:      remove_aresta_congestionada ( $G, s_k, D^k, t^k$ )
12:    fim se
13:     $T_k \leftarrow$  constroi_arvore ( $G, s_k, D^k, t^k$ )
14:     $T' \leftarrow T' \cup T_k$ 
15:  fim se
16:  atualiza_capacidade_arestas ( $G, t^k, T$ )
17: fim para
```

O fluxo de execução do Algoritmo 22 quando aplicado a construção de solução tem as seguintes definições: BL e BT são listas vazias e $iter = 0$. A lista BT vazia na primeira iteração significa que todas as árvores devem ser construídas. A solução T também é vazia. O laço das linhas 2-17 é utilizado para criar as $|W|$ árvores que compõem a solução T . Como $BT = \emptyset$, então o algoritmo vai para construção da árvore que compreende as linhas 9-16. Considerando que $iter = 0$, então a lista de arestas BL é inicializada com arestas da solução em construção, T , na linha 7. Observe-se que as arestas de T são consideradas por custo, ordem decrescente, e apenas um percentual ϑ é escolhida para remoção. A remoção é realizada na linha 9 e garante que o grafo não fique desconectado para o grupo k . Por fim, a árvore é construída na linha 13 por meio do algoritmo de caminho mais curto. A capacidade de cada aresta do grafo é atualizada na linha 16.

Por outro lado, quando o Algoritmo 22 é utilizado para pertubação, o cenário é diferente. As listas BL e BT são entradas do algoritmo e possuem informações para controlar

a perturbação da solução dada como entrada. $iter \geq 1$ e a solução T não é vazia. No caso da perturbação da solução há duas possibilidades: ou uma árvore é mantida na solução, ou ela é removida e uma nova árvore a substitui. O laço das linhas 2-17 é utilizado para perturbar a solução. Na linha 3, é verificado se a árvore T_k deve ser mantida na solução. Em caso positivo, ela é adicionada à solução na linha 4. Se a árvore T_k não vai ser reutilizada, então o algoritmo executa a linha 9 que remove arestas do grafo. Neste caso, BL é não-vazia e contém arestas ordenadas por custo decrescente. Em seguida, uma nova remoção de arestas é realizada na linha 11. Esta remoção tem por objetivo controlar a degradação do valor de capacidade residual. Neste momento, uma nova árvore é criada e adicionada a solução, linhas 13 e 14. A capacidade de cada aresta do grafo é atualizada na linha 16.

6.1.5.2 Busca Local

O operador de busca local do ILS utiliza os Algoritmos 10 e 11. A ideia é aplicá-lo sequencialmente para reduzir o custo e aumentar a capacidade residual fazendo bloqueio de arestas durante o processo. O procedimento é ilustrado no Algoritmo 23 e recebe como parâmetros o grafo do problema, G , o conjunto de grupos *multicast*, W , uma solução, T , e ϖ que é utilizado no procedimento de remoção de arestas.

Algoritmo 23 Algoritmo de Busca Local do ILS

Entrada: G, W, T, ϖ

Saída: $S^* = \{T_1, T_2, \dots, T_{|W|}\}$

- 1: $S' \leftarrow \text{refinamento_por_custo}(G, W, S)$ ▷ Algoritmo 10
 - 2: **para** $T_k \in S'$ **faça**
 - 3: $\text{criar_remover_ciclos}(T_k, S', t^k, D^k)$ ▷ Algoritmo 11
 - 4: **fim para**
 - 5: $\text{remove_arestas}(G, \varpi)$
 - 6: $S^* \leftarrow \text{refinamento_por_Z}(G, W, S')$ ▷ Algoritmo 10
 - 7: Repete linhas uma vez mais 1-5.
-

O Algoritmo 23 aplica uma sequência de operações de busca local visando diminuir o custo e aumentar a capacidade residual da solução. Na linha 1, o Algoritmo 10 é aplicado para melhorar o custo da solução. Em seguida, o Algoritmo 11 é aplicado para criar e remover ciclos com o objetivo de reduzir o custo da solução, linha 3. Neste momento, um percentual ϖ de arestas de G é removido. Estas arestas são ordenadas de modo decrescente por valor de custo. Isso é realizado para evitar que durante o aumento de Z , o valor de custo tenha alta degradação (considerando que a aresta pode voltar para solução). Na linha 6, aplica-se o Algoritmo 11 para melhoria do valor de Z . O processo é repetido novamente e o algoritmo encerra.

6.1.5.3 Algoritmo Proposto

No Algoritmo 24, o processo de busca considerando a abordagem ILS é apresentado. Os seguintes parâmetros de entrada são necessários: grafo do problema, G , conjunto de grupos *multicast*, W , ϑ utilizado no algoritmo de construção de solução, ϖ utilizado no algoritmo de busca local, upd que é utilizado no algoritmo para determinar quando a lista BT deve ser atualizada e max_iter que é o limite máximo de iterações.

Algoritmo 24 Algorithm ILS proposto.

Entrada: $G, W, \vartheta, \varpi, upd, max_iter$ **Saída:** $T^{best} = \{T_1, T_2, \dots, T_{|W|}\}$

```

1:  $BL \leftarrow \emptyset$ 
2:  $BT \leftarrow \emptyset$ ;
3:  $c \leftarrow 1$ 
4:  $S \leftarrow \text{construcao\_solucao}(G, W, \vartheta, BL, BT, 0)$  ▷ Algoritmo 22.
5:  $T^{best} \leftarrow \text{busca\_local}(S, G, W, \varpi)$  ▷ Algoritmo 23.
6:  $\text{atualiza\_BT}(T^{best}, BT)$ 
7: para  $i \leftarrow 1$  to  $max\_iter$  faça
8:    $\text{atualiza\_BL}(BL, T^{best}, \vartheta)$ 
9:    $S \leftarrow \text{construcao\_solucao}(G, W, \vartheta, T^{best}, BL, BT, i)$  ▷ Segundo Fluxo -
   perturbação.
10:   $T' \leftarrow \text{busca\_local}(S, G, W, \varpi)$ 
11:   $T^{best} \leftarrow \text{atualiza\_melhor}(T', T)$ 
12:  se  $c = upd$  então
13:     $c \leftarrow 1$ 
14:     $\text{atualiza\_BT}(T^{best}, BT)$ 
15:  fim se
16:   $c = c + 1$ 
17: fim para
```

O Algoritmo 24 inicia o processo de otimização definindo as listas BL e BT como vazias nas linhas 1 e 2. Em seguida, a variável contadora, c , é definida na linha 3 (esta variável é utilizada para contabilizar o número de iterações que BT não é atualizada). Uma solução inicial é criada na linha 4, operador de busca local é aplicado na linha 5 e a lista BT é definida na linha 6. Neste momento, o laço das linhas 7-17 é utilizado para evoluir a solução inicial. Na linha 8, a lista de arestas bloqueadas por custo, BL , é atualizada. Na linha 9, uma nova solução é obtida pela perturbação da solução T^{best} . Note-se que o algoritmo de construção de solução é aplicado seguindo seu segundo fluxo de execução. A busca local é aplicada a S gerando T' na linha 10. Em seguida, a melhor solução é atualizada na linha 11. Ao final da iteração, verifica-se a possibilidade de atualizar a lista BT . Isso é realizado quando $c = upd$.

6.2 Heurística MMVD

Nesta seção, um algoritmo heurístico é proposto para o problema de roteamento *multicast* multissessão com múltiplas fontes por grupo. Este modelo foi proposto por Chen et al. (2013). O modelo foi analisado na Seção 3.4.2.2 do Capítulo 3. O algoritmo proposto obteve êxito em comparações com o trabalho de Chen et al. (2013) e Guo et al. (2017). A ideia do algoritmo é utilizar os diagramas de Voronoi para obter de forma eficiente as árvores de distribuição que compõem a solução final.

6.2.1 Diagramas de Voronoi

O diagrama de Voronoi consiste de uma partição de um plano em regiões com base na distância entre pontos presentes no mesmo. Cada região que está presente na partição possui um ponto gerador (chamado também de sítio) e é obtida conectando os demais pontos do plano aos pontos geradores com base em métricas como a distância euclidiana, mais detalhes em (AURENHAMMER, 1991).

Um digrama de Voronoi pode ser definido formalmente da seguinte forma. Considere um espaço métrico S e o subconjunto de S definido como $I = \{I_1, I_2, \dots, I_n\}$. O conjunto I define os pontos geradores do diagrama. Associado ao espaço S , define-se uma função de distância, f . Além disso, considere o conjunto $R = \{R_1, R_2, \dots, R_m\}$ que determina a m -ésima região do diagrama relacionada ao m -ésimo ponto gerador.

Uma região R_i é definida pela Equação 6.3:

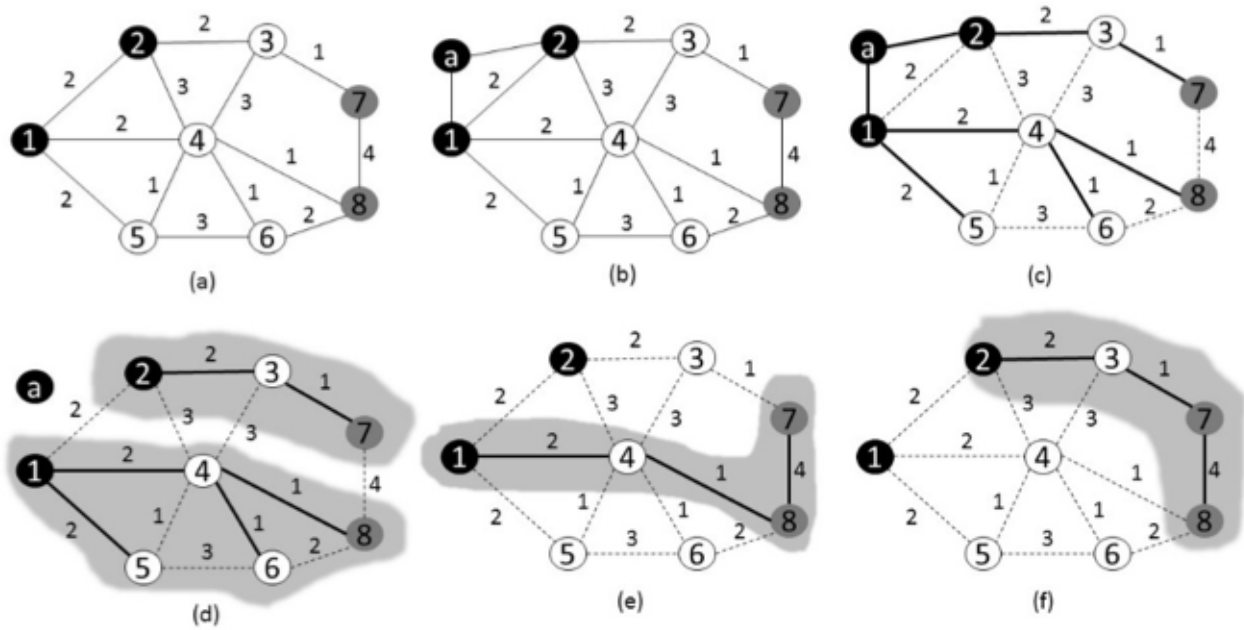
$$R_i = \{x \in S | f(x, I_i) \leq f(x, I_z) \forall z \neq i\} \quad (6.3)$$

Utilizando o espaço Euclidiano, a função f determina a distância entre um ponto $x \in S$ e qualquer ponto gerador do conjunto I .

É possível considerar uma generalização dos diagramas de Voronoi em grafos por meio do seu particionamento. Por exemplo, é possível definir m regiões em um grafo G definindo $R_j \subseteq V, 1 \leq j \leq m$. O diagrama de Voronoi obtido de V diz respeito a R_j . O diagrama de Voronoi obtido da partição aplicada a V diz respeito ao conjunto de geradores I . Onde há m partições de modo que: $V_j \cap V_l = \emptyset$ e $V_j \cup V_l = V, j \neq l, 1 \leq j \leq l \leq m$.

Considerando a generalização acima, pode-se utilizar diferentes métricas para construção do diagrama de Voronoi tomando um subconjunto de vértices de V . Por exemplo, pode-se construir o diagrama tomando o caminho mais curto ou utilizando o número de *hops* como métrica.

Para ilustrar a construção de um diagrama de Voronoi considere o caminho mais curto como métrica. Além disso, considere um grafo G' que é obtido do grafo do problema, $G = (E, V)$. O grafo G' é obtido através da adição de um vértice artificial a . Este vértice será conectado aos vértices geradores, $I_j, 1 \leq j \leq m$. Arestas (a, I_j) são adicionadas

Figura 19 – Exemplo de criação de diagrama de Voronoi em grafos.

Fonte: própria.

ao conjunto de arestas E . O próximo passo após a criação do grafo G' é a aplicação do algoritmo de caminho mais curto iniciando no vértice a . Em seguida, o vértice a e todas as arestas conectadas a ele no grafo G' .

Na Figura 19, um exemplo da construção do diagrama de Voronoi é ilustrado com o caminho mais curto como métrica.

O exemplo ilustrado na Figura 19 tem o conjunto de vértices $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$, o conjunto de geradores $I = \{1, 2\}$. O grafo do problema, G , é ilustrado na Figura 19 (a). Na Figura 19 (b), ilustra-se o grafo G' obtido através da adição do vértice a , criando as arestas $(a, 1)$ e $(a, 2)$. Em seguida, o algoritmo de caminho mais curto é aplicado ao grafo G' , obtendo a árvore enraizada em a na Figura 19 (c). Removendo-se o vértice a e suas arestas, obtém-se as duas árvores conforme ilustrado na Figura 19 (d). Observe-se que duas regiões foram criadas $R_1 = \{1, 5, 4, 6, 8\}$ e $R_2 = \{2, 3, 7\}$.

Este exemplo ilustra uma maneira simples de construir uma solução para o problema ilustrado na Seção 3.4.2.2 do Capítulo 3. Por exemplo, considere que os vértices 1 e 2 são as fontes de transmissão e os vértices 7 e 8 são vértices clientes. Conforme ilustrado na Figura 19 (d), o custo de atendimento aos vértices 7 e 8 é igual a 6. Este valor é melhor que os valores obtidos utilizando apenas uma fonte, seja o vértice 1 ou vértice 2, conforme ilustrado nas Figuras 19 (e) e (f).

6.2.2 Algoritmos Proposto

A proposta de uso de diagrama de Voronoi para o problema proposto por Chen et al. (2013) visa a construção eficiente de árvores de distribuição considerando a abordagem ilustrada na Figura 19.

O Algoritmo 25 é utilizado para construção de soluções para o problema mencionado. Ele recebe como entrada o grafo de problema, G , e o conjunto de *streams multicast*, W . Considere que W tem exatamente a mesma interpretação adotada em outros problemas, ou seja, representa o conjunto de grupos *multicast*. Além disso, considere que neste problema $|S_k| \geq 1$.

Algoritmo 25 Heurística Multi-Stream Multi-Source Voronoi Diagram - MMVD

Entrada: Grafo G , *Streams* W

```

1: para  $w_k \in W$  faça
2:    $R_k \leftarrow S_k$ 
3:    $Forrest_k \leftarrow \text{construir\_floresta}(D^k, R_k, t^k, G)$ 
4:    $E_k \leftarrow \text{arestas}(Forrest_k)$ 
5:   para  $(i, j) \in E_k$  faça
6:      $b_{(i,j)} \leftarrow b_{(i,j)} - t^k$ 
7:   fim para
8: fim para

```

O MMVD inicia o processo de otimização definindo o conjunto $R_k = S_k$ na linha 2. Em seguida, uma floresta é construída associada ao k -ésimo grupo na linha 3. As arestas utilizadas na construção de $Forrest_k$ têm sua capacidade atualizada na linha 6. O processo de construção utilizado é o mesmo que aquele apresentado na Figura 19. Entretanto, a construção é baseada em um algoritmo de Dijkstra modificado para construção de uma *widest path tree*, que é definida como árvore com caminhos com maior capacidade possível que conectem S_k e D^k . As modificações no algoritmo de Dijkstra são as seguintes. A criação das arestas ligando a a vértices do grafo considerando capacidade residual infinita. Além disso, a regra de atualização de distância original do algoritmo de Dijkstra é substituída pela regra definida na Equação 6.4:

$$widto(x) = \max_{e=(v,x) | v \in T} (\min(widto(v), wid(e))) \quad (6.4)$$

Esta regra é utilizada para determinar a distância de um vértice x para o vértice artificial a . A função $widto(v)$ é utilizada para determinar o valor da capacidade mínima no caminho entre a e v . Já a função $wid(e)$ é utilizada para determinar a capacidade residual da aresta (v, x) . Portanto, o valor associado a cada vértice é o gargalo no caminho de a até x . O vértice v escolhido é aquele que vai maximizar o valor da capacidade residual do caminho até x .

6.2.3 Análise de Complexidade

A análise de complexidade do algoritmo será realizada considerando a seguinte notação: $m = |E|$, $n = |V|$, $r = |W|$, $s = |S_k|$ e $d = |D^k|$. Além disso, os conjuntos S_k e D^k têm a mesma cardinalidade para todo grupo *multicast*.

O Algoritmo MMVD tem sua complexidade definida pelo procedimento de construção de floresta. O algoritmo modifica o grafo inserindo arestas adicionais cuja complexidade é $\Theta(s)$. Em seguida, utiliza-se o algoritmo modificado de Dijkstra cuja complexidade é $O(m + n \log n)$. Por fim, ele atualiza as arestas do grafo com relação a capacidade que tem complexidade $O(m)$. A complexidade de construção de uma floresta é $O(s + m + n \log n)$. Esse procedimento é realizado r vezes. Portanto, a complexidade geral do algoritmo é $O(rs + rm + n \log n)$.

O algoritmo proposto por Chen et al. (2013) também tem sua complexidade definida pelo algoritmo de Dijkstra. Entretanto, este procedimento é aplicado s vezes por grupo *multicast* em oposição ao MMVD que executa apenas uma vez. Além disso, há uma segunda fase que visa a construção da floresta para o grupo $k = 1, \dots, r$. Nesta fase, um caminho entre um vértice de D^k até um vértice fonte é construído. Isso garante que cada vértice cliente seja conectado apenas a uma fonte e com máxima capacidade no caminho escolhido. A complexidade geral da segunda fase $O(nds)$. Portanto, a complexidade final do algoritmo é $O(rsm + rsn \log n + rnds)$.

O algoritmo proposto por Guo et al. (2017) tem sua complexidade também orientada por construção de caminho mais curto. A primeira fase do algoritmo consiste em construir $(s + d)^2$ caminhos mais curtos para gerar um grafo completo, G^c . Logo, a complexidade da primeira fase é $O((s + d)^2 \cdot (m + n \log n))$. A segunda fase do procedimento deve encontrar uma árvore geradora, T , no grafo G^c cuja complexidade é $O((|s| + |d|)^2)$. Logo, a complexidade total do procedimento é: $O((|s| + |d|)^2 + (s + d)^2 \cdot (m + n \log n))$. Vale ressaltar que este trabalho considera apenas uma árvore de distribuição a ser construída. A adaptação do algoritmo para o problema de múltiplas sessões aumenta a complexidade.

Como pode ser visto nas análises acima, o algoritmo MMVD possui melhor complexidade em relação aos demais procedimentos. Isso também é verificado experimentalmente conforme será ilustrado no Capítulo 7

6.3 Algoritmos para Otimização Multiobjetivo

Nesta seção, a representação de solução adotada para soluções do modelo multiobjetivo proposto no Capítulo 5 é apresentada, Seção 6.3.1. Em seguida, apresentam-se os operadores de criação de solução utilizados nos algoritmos multiobjetivo e também no algoritmo Transgenético proposto, Seção 6.3.2. Por fim, os algoritmos de recombinação e mutação são apresentados na Seção 6.3.3.

6.3.1 Representação de Solução

A representação de solução adotada consiste em uma lista de árvores. Cada árvore consiste em uma lista de caminhos entre cada par fonte-cliente. Esta forma de representação permite manipulações a nível de árvore e caminhos de forma eficiente contribuindo para complexidade de tempo. Mais detalhes desta representação estão presentes na Seção 6.1.4.1.

6.3.2 Operador de Criação de Solução

O operador de criação de soluções emprega o uso de algoritmos de construção de árvores de distribuição. Três formas diferentes de criação de árvores foram utilizadas.

A primeira forma visa a otimização do objetivo de custo da solução através de árvores enraizadas no vértice fonte utilizando o algoritmo de Dijkstra. A segunda forma de construção visa a otimização da capacidade residual, Z , através do algoritmo de Dijkstra adaptado conforme utilizado no algoritmo MMVD, Seção 6.2. Por fim, o algoritmo de busca em profundidade é aplicado para construção de árvores que contenham caminhos com baixo número de *hops* entre fonte e vértice de destino.

O Algoritmo 26 recebe como entrada o grafo do problema, G , o conjunto de grupos *multicast*, W e o tipo de operador utilizado na construção das árvores de distribuição, H .

Algoritmo 26 Algoritmo de criação de solução.

Entrada: Grafo G , Conjunto de Grupos W , Tipo H

Saída: $T = \{T_1, T_2, \dots, T_{|W|}\}$

```

1:  $T \leftarrow \emptyset$ 
2: para  $w_k \in W$  faça
3:    $T_k \leftarrow \text{construir\_arvore}(S_k, D^k, G, H)$ 
4:    $T \leftarrow T \cup T_k$ 
5:   atualizar_arestas_grafo( $G, t^k$ )
6: fim para
```

O processo de criação de soluções ilustrado no Algoritmo 26 depende dos operadores de construção de solução, que é determinado pelo parâmetro $H \in \{1 : \text{custo}, 2 : Z, 3 : \text{hop}\}$. Na linha 1, o algoritmo define uma lista de árvores vazia, T . Em seguida, o laço das linhas 2-6 é utilizado para construir as $|W|$ árvores de distribuição. Na linha 3, uma árvore é construída considerando um dos tipos de operadores ($\{1, 2, 3\}$). Em seguida, na linha 4 a árvore T_k é incorporada a solução T . O grafo é atualizado para registrar a nova capacidade de suas arestas. Observe-se que o algoritmo de construção de árvores, **construir_arvores()**, opera da seguinte forma para garantir diversidade na inicialização das soluções: a fonte da árvore construída é sempre escolhida de modo aleatório considerando os vértices do conjunto $D^k \cup S_k$. Isso garante diversidade na construção de árvores independentemente do operador utilizado.

6.3.3 Operadores de Recombinação e Mutação

Nesta seção, os operadores de recombinação e mutação são definidos. Quatro operadores de recombinação foram implementados e possuem propósitos diferentes dentro do processo de busca. Alguns deles são aplicados a árvore e também aos caminhos.

6.3.3.1 Recombinação por Árvore

O Algoritmo de recombinação por árvore utilizado nos algoritmos para o problema multiobjetivo é o Algoritmo 19. O processo de execução ocorre da mesma forma que no Algoritmo genético 21. Duas soluções são passadas como parâmetro. O cromossomo é dividido em uma posição gerada aleatoriamente e uma nova solução é criada obtendo informações das soluções dadas como entrada.

6.3.3.2 Recombinação por Custo

O Algoritmo 27 é utilizado para fazer a criação de um novo indivíduo apenas com árvores de baixo custo. Ele recebe como parâmetros duas soluções, T^1 e T^2 , e o conjunto de grupos, W . O processamento inicia com a definição de uma solução, T , vazia na linha 1. Em seguida, o laço das linhas 2-8 é utilizado para transferir as árvores de menor custo para nova solução. Na linha 4, a árvore de T^1 é repassada por ter menor custo e, caso contrário, na linha 6 a árvore de T^2 é repassada.

Algoritmo 27 Algoritmo de recombinação com base no custo de árvore.

Entrada: Soluções T^1, T^2 e Grupos W

Entrada: T

```
1:  $T \leftarrow \emptyset$ 
2: para  $k \in W$  faça
3:   se custo ( $T_k^1$ ) < custo ( $T_k^2$ ) então
4:      $T \leftarrow T \cup T_k^1$ 
5:   senão
6:      $T \leftarrow T \cup T_k^2$ 
7:   fim se
8: fim para
```

6.3.3.3 Recombinação por Caminho

O Algoritmo 28 é utilizado para realizar recombinação fazendo manipulação de caminhos de uma árvore. Ele recebe como parâmetros duas soluções, T^1 e T^2 , e o conjunto de grupos, W . A árvore T^1 é dominante e passará a maior parte das informações para a nova solução. Apenas uma das árvores será modificada sendo composta por caminhos de T^1 e T^2 .

Algoritmo 28 Algoritmo de recombinação por caminho.**Entrada:** Soluções T^1, T^2 e Grupos W **Saída:** T

```

1:  $k \leftarrow \text{valor\_aleatorio}(1, |W|)$ 
2:  $T \leftarrow \emptyset$ 
3: para  $i = 1$  até  $k - 1$  faça
4:    $T \leftarrow T \cup T_i^1$ 
5: fim para
6: para  $i = k + 1$  até  $|W|$  faça
7:    $T \leftarrow T \cup T_i^1$ 
8: fim para
9:  $T_k \leftarrow \emptyset$ 
10:  $D \leftarrow |D^k|$ 
11: para  $p = 1$  até  $D/2$  faça
12:    $T_k \leftarrow T_k \cup T_{kp}^1$ 
13: fim para
14: para  $p = (D + 1)$  até  $D$  faça
15:    $T_k \leftarrow T_k \cup T_{kp}^2$ 
16: fim para
17: consolidar  $(T_k)$ 
18:  $T \leftarrow T \cup T_k$ 

```

A recombinação por caminho inicia pela definição da árvore, índice k , que será modificada através da mistura de caminhos de duas soluções na linha 1. Em seguida, uma solução vazia, T , é definida na linha 2. Os laços das linhas 3-5 e 6-8 são utilizados para passar as árvores de T^1 para a solução T . Observe-se que a árvore associada ao índice k não é repassada. Na linha 9, a árvore T_k é definida como vazia e o número de caminhos da fonte até D^k é definido na linha 10. Os laços das linhas 11-13 e 14-16 são utilizados para passar os caminhos de T^1 e T^2 para a solução T nas linhas 12 e 15, respectivamente. A árvore T_k é corrigida para evitar ciclos na linha 17 e adicionada à solução na linha 18.

6.3.3.4 Recombinação por *Hops*

O Algoritmo 29 é utilizado para recombinação misturando caminhos das árvores que compõem uma solução. O objetivo é construir árvores que utilizem o menor caminho possível da fonte para cada vértice cliente. Ele recebe duas soluções, T^1 e T^2 , e o conjunto de grupos, W .

A construção de uma nova solução inicia com a definição de uma solução vazia na linha 1. Em seguida, o laço das linhas 2-14 é utilizado para fazer a construção da solução T . Para cada grupo *multicast*, uma nova árvore será criada no laço das linhas 5-11. O primeiro passo é definir o número de caminhos da árvore na linha 3. Em seguida, a árvore T_k é definida como vazia na linha 4. Os D caminhos são inseridos na árvore T_k . Para cada caminho p , verificar qual árvore tem o menor caminho na linha 6. Se a árvore de

Algoritmo 29 Algoritmo de recombinação por hops.

Entrada: Soluções T^1, T^2 e Grupos W **Saída:** T

```
1:  $T \leftarrow \emptyset$ 
2: para  $k \in W$  faça
3:    $D \leftarrow |D^k|$ 
4:    $T_k \leftarrow \emptyset$ 
5:   para  $p = 1$  até  $D$  faça
6:     se  $\text{hop}(T_{kp}^1) < \text{hop}(T_{kp}^2)$  então
7:        $T_{kp} \leftarrow T_{kp}^1$ 
8:     senão
9:        $T_{kp} \leftarrow T_{kp}^2$ 
10:    fim se
11:  fim para
12:  consolidar ( $T_k$ )
13:   $T \leftarrow T \cup T_k$ 
14: fim para
```

T^1 tiver o menor caminho, este é cedido na linha 7. Caso contrário, o caminho de T^2 é repassado na linha 9. Ao final da construção, a árvore T_k é consolidada na linha 12 para evitar ciclos. Por fim, a árvore é adicionada à solução T na linha 13.

6.3.3.5 Recombinação diferencial

O operador de recombinação diferencial utiliza a ideia presente na abordagem GDE, onde se considera o uso de mais de dois indivíduos para fazer cruzamento e gerar um novo indivíduo. As manipulações realizadas neste operador são consideradas dentro da representação de soluções já utilizadas (definida na Seção 6.3.1), diferentemente do que ocorre na abordagem original.

O Algoritmo 30 tem como ideia central o uso de três indivíduos. O novo indivíduo será gerado pela mistura de informações. Este processo ocorre passando uma árvore, T_k , de um dos três indivíduos para a nova solução, ou criando uma nova árvore através dos caminhos das árvores de dois indivíduos selecionados aleatoriamente. O critério de escolha do caminho é o menor custo. O procedimento recebe como entrada as soluções, T^1, T^2 e T^3 , e o conjunto de grupos *multicast*, W .

O processo de criação de um novo indivíduo através do operador ilustrado no Algoritmo 30 é realizada da seguinte forma. Na linha 1, define-se um solução vazia, T . Em seguida, o laço das linhas 2-22 é executado onde uma árvore é construída para cada grupo *multicast* k . Cada árvore T_k da solução pode ser obtida de duas maneiras. O primeiro método consiste simplesmente em repassar a árvore T_k de um dos indivíduos sorteados aleatoriamente, linhas 5 e 6. Por outro lado, o segundo método consiste em construir uma árvore T_k utilizando caminhos das soluções T^1, T^2 ou T^3 . Isso é realizado da seguinte forma. Na linha 8, define-se o número de caminhos da árvore T_k . Na linha 9, cria-se a árvore T_k vazia.

Algoritmo 30 Algoritmo de recombinação diferencial.**Entrada:** Soluções T^1, T^2 e T^3 , Grupos W .**Saída:** T

```

1:  $T \leftarrow \emptyset$ 
2: para  $k = 1$  até  $|W|$  faça
3:    $Metodo \leftarrow \text{sorteio\_aleatorio}(\{1, 2\})$ 
4:   se  $Metodo = 1$  então
5:      $j \leftarrow \text{escolhe\_individuo}(\{1, 2, 3\})$ 
6:      $T_k \leftarrow T_k^j$ 
7:   senão
8:      $D \leftarrow |D^k|$ 
9:      $T_k \leftarrow \emptyset$ 
10:    para  $p = 1$  até  $D$  faça
11:       $l \leftarrow \text{escolhe\_individuo}(\{1, 2, 3\})$ 
12:       $m \leftarrow \text{escolhe\_individuo}(\{1, 2, 3\} \setminus l)$ 
13:      se  $\text{custo}(T_{kp}^l) < \text{custo}(T_{kp}^m)$  então
14:         $T_{kp} \leftarrow T_{kp}^l$ 
15:      senão
16:         $T_{kp} \leftarrow T_{kp}^m$ 
17:      fim se
18:    fim para
19:    consolidar ( $T_k$ )
20:  fim se
21:   $T \leftarrow T \cup T_k$ 
22: fim para

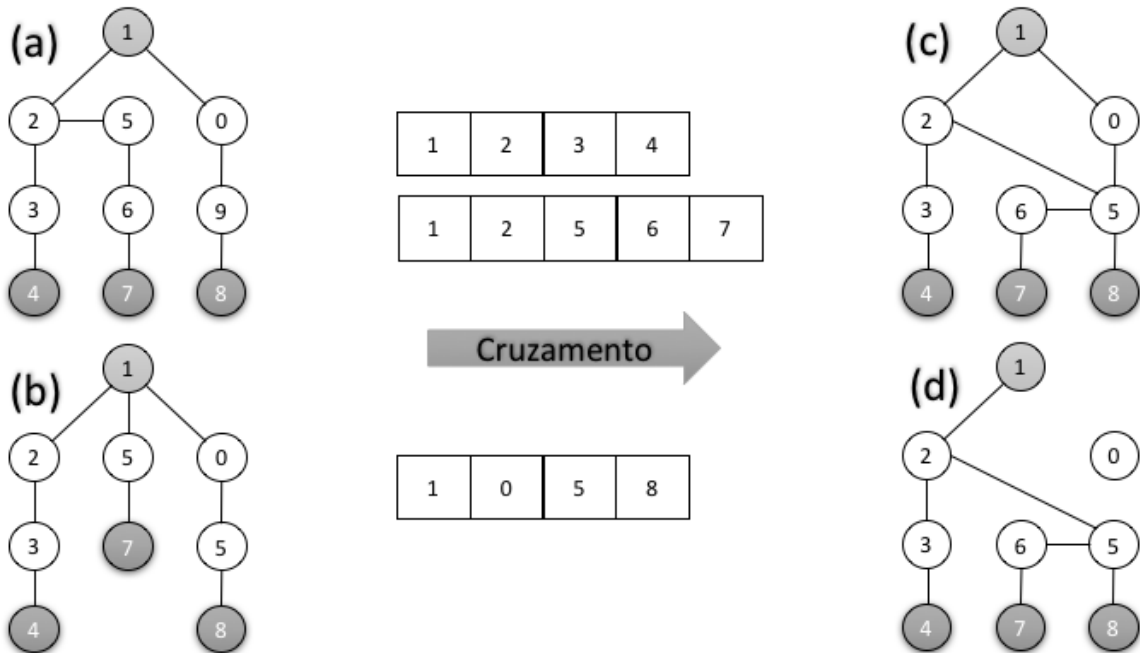
```

O laço das linhas 10-18 é utilizado para transferir os caminhos para a nova árvore. Dois indivíduos são escolhidos aleatoriamente nas linhas 11 e 12. Se o caminho p do indivíduo l na árvore k for menos caro que aquele do indivíduo m , então T^l cede o caminho na linha 14. Caso contrário, T^m cede o caminho na linha 16. Por fim, a árvore é consolidada na linha 19. Ao final da construção da k -ésima árvore, a mesma é adicionada à solução T na linha 21.

6.3.3.6 Consolidação de Árvore

Os Algoritmos 28, 29 e 30 aplicam o procedimento **consolidar** às árvores construídas por combinação de caminhos. O processo de consolidação de caminho dá-se conforme ilustrado na Figura 20.

Considere duas árvores da Figura 20 (a) e Figura 20 (b) associadas ao grupo *multicast* k ($D^k = \{4, 7, 8\}$ e $S_k = \{1\}$), denominadas árvore **A** e árvore **B**, respectivamente. O cruzamento das duas árvores forma uma árvore ilustrada na Figura 20 (c). A nova árvore em formação envolve dois caminhos da árvore A e um caminho da árvore B. O resultado do cruzamento ilustrado na Figura 20 (c) contém ciclo e deve ser removido. O processo de remoção de ciclos considera a ordem dos caminhos na estrutura da solução. O caminho

Figura 20 – Exemplo do procedimento consolidar árvore.

Fonte: própria.

que leva até o vértice 4 é incorporado à árvore. Em seguida, o vértice 7 é adicionado. O processo de adição inicia no último vértice, 7, e vai até o vértice 1 ou um vértice ligado a 1. No exemplo o vértice em questão é o vértice 2. Isso incorpora o caminho 1 – 2 – 5 – 6 – 7. O último vértice, 8, é adicionado à árvore do mesmo modo. Neste caso, ele é conectado a árvore por meio do vértice 5. O resultado é ilustrado na Figura 20 (d). O vértice 0 e suas arestas são desconsiderados.

6.4 Algoritmo Transgenético

Nesta seção, o algoritmo transgenético proposto e seus operadores, plasmídios e transposon são apresentados. Dois operadores plasmídio foram projetados e o transposon consiste em aplicar um algoritmo de busca local.

6.4.1 Informação Genética

Os algoritmos transgenéticos beneficiam-se de informações obtidas antes e durante o processo de otimização. Essas informações são cruciais para o bom desempenho da abordagem já registrado na literatura. Neste trabalho, as informações utilizadas pelos vetores transgenéticos são caminhos entre o vértice fonte e um vértice cliente do k -ésimo grupo

multicast. Além disso, também são mantidas informações sobre as árvores de menor custo para cada grupo.

Os operadores definidos nos Algoritmos 31 e 32 utilizam a base de informações associadas a caminhos e árvores, respectivamente. A lista ilustra a composição das bases de informação:

- **Base 1** - São armazenados K –caminhos de menor custo entre um par de vértice $s, d | s \in S_k, d \in D^k, \forall k \in W$;
- **Base 2** - São armazenados K –caminhos de menor comprimento entre um par de vértice $s, d | s \in S_k, d \in D^k, \forall k \in W$;
- **Base 3** - Os caminhos de soluções de elite são armazenados para uso na iteração corrente. Um caminho para cada par de vértice $s, d | s \in S_k, d \in D^k, \forall k \in W$;
- **Base 4** - As árvores de menor custo são armazenadas em uma solução $T^{best, cost}$.

As bases de informação 1, 2 e 3 são utilizadas para modificar árvores. O algoritmo proposto Yen (1971) é utilizado para obter as bases 1 e 2. Os caminhos de elite da base 3 são obtidos a partir das soluções de elite (definidas na Seção 6.4.3). Por fim, a base 4 é composta por árvores de menor custo para cada grupo. Elas são atualizadas durante todo o processo de busca.

6.4.2 Plasmídios e Transposon

Os operadores plasmídio propostos atuam na alteração das soluções por meio da manipulação de caminhos e também árvores. Os Algoritmos 31 e 32 são utilizados para tais manipulações, respectivamente.

Plasmídio

O Algoritmo 31 atua na manipulação de soluções através da inserção de caminhos na solução. Os caminhos inseridos são classificados por custo ou *hop*. Para tanto, utiliza-se a base de informações genéticas definidas na Seção 6.4.1. Ele recebe como parâmetros uma solução, T , o conjunto de grupos *multicast*, W , e a base de informações, *host*.

O processo de inserção de informações na solução vai gerar um indivíduo T' . O laço das linhas 2-8 é utilizado para percorrer o conjunto de árvores da solução. Na linha 3, um vértice de destino do grupo k é escolhido aleatoriamente. Em seguida, um caminho da fonte do grupo, S_k , para o vértice d é escolhido aleatoriamente e armazenado em P_d , linha 4. Este caminho é inserido na árvore substituindo o antigo caminho na linha 5. Em seguida, a árvore T_{aux} é consolidada na linha 6 conforme a Figura 20. Por fim, ela é adicionada a solução na linha 7.

Algoritmo 31 Algoritmo **PathPlasmídio**.

Entrada: Solução T , Grupos W , $host$ **Saída:** T'

```
1:  $T' \leftarrow \emptyset$ 
2: para  $k \in W$  faça
3:    $d \leftarrow \text{aleatorio}(D^k)$ 
4:    $P_d \leftarrow \text{escolhe\_caminho\_aleatorio}(host, d)$ 
5:    $T_{aux} \leftarrow \text{substituir}(T_k, P_d)$ 
6:    $\text{consolidar}(T_{aux})$ 
7:    $T' \leftarrow T \cup T_{aux}$ 
8: fim para
```

O Algoritmo 32 é utilizado para manipular soluções inserido árvores com o menor custo possível durante a busca. As árvores inseridas constituem o banco de informações descrito na seção anterior. Os parâmetros dados como entrada são: uma solução, T ; conjunto de grupos *multicast*; a melhor solução, $T^{best, cost}$, com relação ao custo de configuração das árvores e uma taxa de aplicação da inserção, P_{rate} .

Algoritmo 32 Algoritmo **TreePlasmídio**.

Entrada: Solução T , W , $T^{best, cost}$, P_{rate} **Saída:** T'

```
1:  $T' \leftarrow \emptyset$ 
2: para  $k \in W$  faça
3:    $p \leftarrow \text{aleatorio}(0, 1)$ 
4:   se  $p < P_{rate}$  então
5:      $T' \leftarrow T' \cup T_k^{best, cost}$ 
6:   senão
7:      $T' \leftarrow T' \cup T_k$ 
8:   fim se
9: fim para
```

A inserção de árvores realizada pelo Algoritmo 32 é realizada no laço das linhas 2-9. Um valor no intervalo x é sorteado uniforme e aleatoriamente na linha 3. Se o valor for menor que P_{rate} , então a árvore T_k será substituída por $T_k^{best, cost}$, que corresponde a árvore de menor custo para o grupo k , linha 5. Caso contrário, a árvore T_k da solução T será passada para nova solução na linha 7.

Transposon

O Algoritmo Transposon utilizado é uma adaptação do procedimento de busca local aplicado em árvores descrito no Algoritmo 11. Este algoritmo é aplicado apenas a um conjunto de soluções denominadas de soluções de elite devido sua complexidade computacional.

Assim sendo, as soluções que forem modificadas pelo Transposon passam a ser soluções de baixo custo. Estas soluções são incorporadas a cada subpopulação. Além disso, suas

informações serão armazenadas em uma base de dados para aplicação em novas soluções por meio de plasmídio.

6.4.3 Subpopulações e Soluções de Elite

Neste trabalho, o algoritmo transgenético proposto evolui uma população dividida em três subpopulações. Além disso, um conjunto de soluções é considerado elite.

Subpopulações

A ideia é ter populações que representam soluções com interesses de otimização distintos. Foram consideradas três subpopulações associadas ao operador de construção de árvores de distribuição descrito na Seção 6.3.2.

Entretanto, o algoritmo proposto é flexível o suficiente para considerar o uso de apenas um tipo de operador em caso de se construir uma população com apenas um algoritmo de construção de árvores. Esse detalhe sobre o operador escolhido é definido por meio de parâmetros.

As populações evoluem separadamente até o fim do número máximo de iterações. Em seguida, são unidas e filtradas com relação a não-dominância.

Soluções de Elite

O conceito de soluções de elite empregado no algoritmo proposto diz respeito a um pequeno número de soluções - fixado a priori. Estas soluções serão consideradas por iteração do algoritmo, ou seja, em cada iteração um conjunto de soluções é selecionado para servir de elite.

Estas soluções, consideradas de elite, passam por processos de melhoria através de um operador específico que será utilizado como transposon. Em seguida, as soluções alimentam a base de informação genética **base 3**. Isso é realizado em todas as iterações do algoritmo.

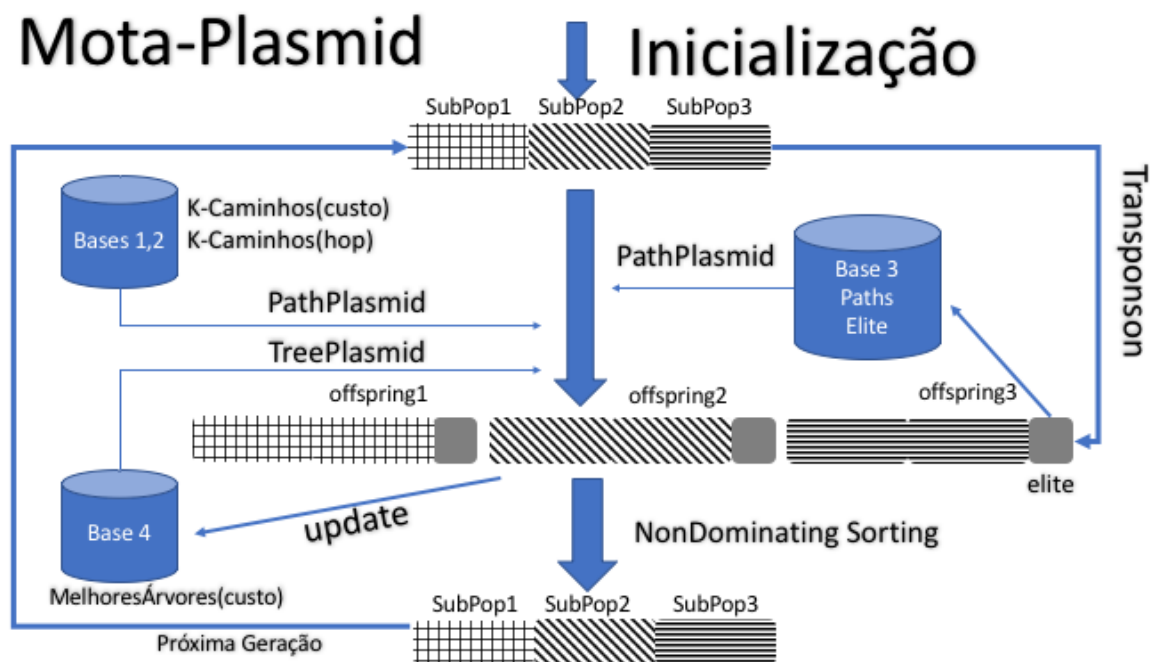
O conjunto de soluções também é separado por subpopulação. Ou seja, cada subpopulação cede um número fixo de indivíduos para servir de elite. Após a criação dos novos indivíduos, todos são unidos para gerar a subpopulação da próxima iteração.

6.4.4 Implementação

O algoritmo transgenético proposto pode ser classificado em duas versões distintas. A primeira versão não faz uso do operador de cruzamento de soluções, apenas operadores plasmídio. A segunda versão incorpora o uso do operador de cruzamento. Essa modificação é realizada por meio de parâmetro de entrada do algoritmo.

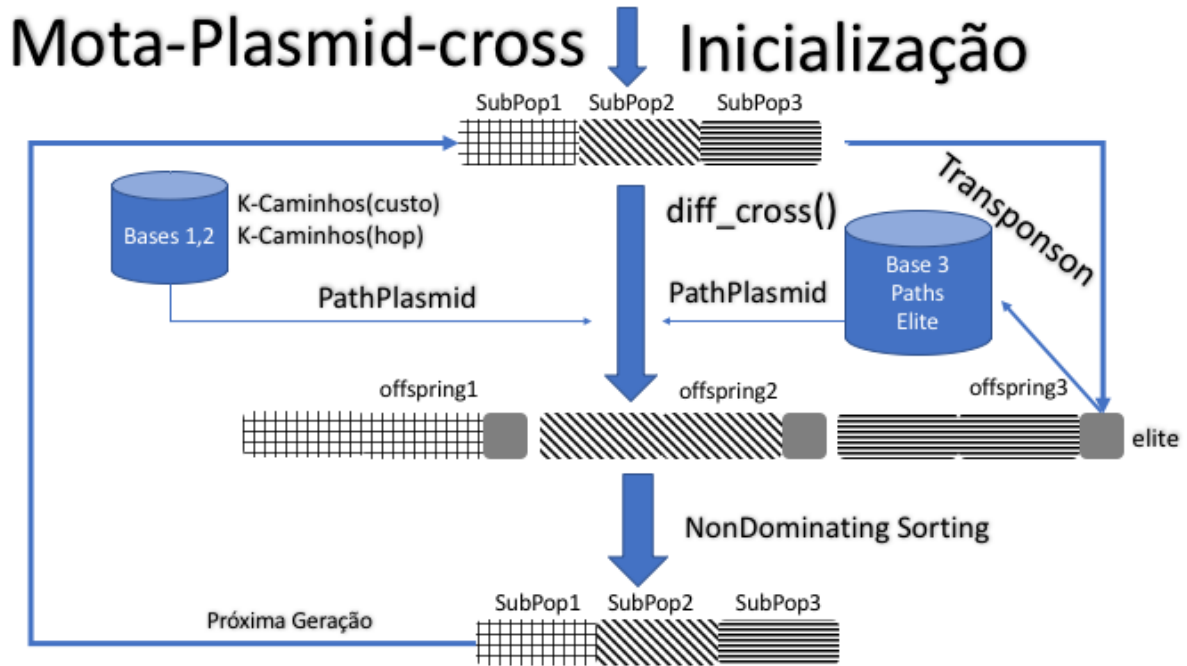
Na Figura 21, o fluxo de execução do algoritmo transgenético sem uso de cruzamento de soluções é apresentado. O primeiro passo é iniciar as subpopulações de indivíduos utilizando operadores diferentes. Em seguida, para cada subpopulação de indivíduos, uma nova população é criada através do uso de plasmídio. Neste caso, dois tipos de plasmídios são utilizados: **PathPlasmidio** e **TreePlasmidio**. Observe-se que uma regra de uso do Plasmídio é violada, pois o mesmo considera que um novo indivíduo deve ser comparado contra aquele do qual foi gerado visando a escolha de apenas um deles. Entretanto, aqui os novos indivíduos são adicionadas as subpopulações. Cada subpopulação possui indivíduos denominados de indivíduos de elite, que são obtidos por operador de transposon. Os indivíduos de elite estão associadas a cada subpopulação. Eles alimentam o banco de informações 3 com caminhos de suas árvores. Assim como o banco de informações 4 é atualizado com as melhores árvores com base no custo de configuração. As populações **offspring*** são filtradas através do algoritmo de classificação e seleção de indivíduos de Deb et al. (2002). Este processo repete-se até um número máximo de iterações.

Figura 21 – Fluxo de Execução do algoritmo transgenético sem uso de cruzamento.



Fonte: própria.

Na Figura 22, o segundo fluxo de execução do algoritmo transgenético é ilustrado. Observe-se que neste caso o cruzamento de soluções é incorporado (na figura é ilustrado como **diff_cross**). Para tanto, o operador utilizado é definido no Algoritmo 30. Além disso, o plasmídio associado as árvores de menor custo, **base 4**, não é utilizado. Portanto, as populações **offspring** são criadas utilizando plasmídio e cruzamento.

Figura 22 – Fluxo de Execução do algoritmo transgenético com cruzamento de soluções.

Fonte: própria.

Os processos de otimização ilustrado nas Figuras 21 e 22 são descritos no Algoritmo 33. Ele recebe como parâmetros o grafo do problema, G , os grupos *multicast*, W , os operadores de construção de árvores, Op , valor booleano que indica se o cruzamento é utilizado, $cross$, a taxa de aplicação de **TreePlasmidio**, P_{rate} e a taxa de aplicação de **PathPlasmidio**, $Plasmid_{rate}$. Ele tem como saída o conjunto de soluções não-dominadas, ND .

O Algoritmo 33 inicia definindo duas bases de informação genética nas linhas 1 e 2. Estas bases são os X -menores caminhos entre a fonte e cada membro do grupo *multicast* para todos os grupos considerando custo e *hop*, respectivamente. Na linha 3, uma base de informação é definida como vazia. Esta base é alimentada pelas soluções de elite durante o processo de evolução. Em seguida, o laço das linhas 4-6 é utilizado para iniciar as subpopulações. Para cada $op \in OP$, uma subpopulação é criada. O procedimento **iniciar_populacao** utiliza o Algoritmo 26 e constrói um número de indivíduos igual a $N/|OP|$.

A partir da definição da população inicial, o laço das linhas 7-35 é utilizado para evoluir a população de indivíduos através do uso de plasmídio e também cruzamento. Entretanto, duas versões do algoritmo serão consideradas com base na utilização e não utilização do cruzamento.

Algoritmo 33 Algoritmo Transgenético Proposto - TA.

Entrada: Grafo G , Grupos W , Operadores Op , $cross$, P_{rate} , $Plasmid_{rate}$, tamanho da população N

Saída: Conjunto de soluções ND

```
1:  $Host_{P,cost} \leftarrow$  iniciar ( $G, W$ )
2:  $Host_{P,hop} \leftarrow$  iniciar ( $G, W$ )
3:  $Host_{P,elite} \leftarrow \emptyset$ 
4: para  $op \in Op$  faça
5:    $Subpop_{op} \leftarrow$  iniciar_populacao ( $G, W, op, N/|OP|$ )
6: fim para
7: enquanto  $i = 1$  até  $max\_iter$  faça
8:    $Elite_i \leftarrow$  prepare_elite ( $Subpop_*$ )
9:    $Host_{P,elite} \leftarrow$  iniciar ( $Elite_i$ )
10:   $T^{best,cost} \leftarrow$  melhor_custo ( $Subpop_*$ )
11:  para  $op \in Op$  faça
12:    para  $j = 1$  até  $|Subpop_{op}|$  faça
13:       $metodo \leftarrow$  aleatorio ( $0, 1$ )
14:      se  $Plasmid_{rate} = metodo$  então
15:         $T^{rand} \leftarrow$  escolhe_individuo ( $Subpop_{op}$ )
16:         $host \leftarrow$  aleatorio ( $Host_{P,cost}, Host_{P,hop}, Host_{P,elite}$ )
17:         $T^j \leftarrow$  plamidio_caminho( $T^{rand}, W, host$ )
18:      senão
19:        se  $cross = TRUE$  então
20:           $parents \leftarrow$  escolhe_tres ( $Subpop_{op}$ )
21:           $T^j \leftarrow$  crossover_diff( $parents, W$ )
22:        senão
23:           $T^{rand} \leftarrow$  escolhe_individuo ( $Subpop_{op}$ )
24:           $T^j \leftarrow$  plamidio_arvore( $T^{rand}, W, T^{best,cost}, P_{rate}$ )
25:      fim se
26:    fim se
27:     $Subpop_{op} \leftarrow Subpop_{op} \cup T^j$ 
28:  fim para
29: fim para
30: para  $op \in Op$  faça
31:    $Subpop_{op} \leftarrow Subpop_{op} \cup Elite^{op}$ 
32:    $Subpop_{op} \leftarrow$  fast_nondominated_sort ( $Subpop_{op}$ )
33: fim para
34:  $i \leftarrow i + 1$ 
35: fim enquanto
36:  $ND \leftarrow \emptyset$ 
37: para  $op \in Op$  faça
38:    $ND \leftarrow ND \cup Subpop_{op}$ 
39: fim para
40:  $ND \leftarrow$  fast_nondominated_sort ( $ND$ )
```

Mota-Plasmid

A versão Mota-Plasmid considera o parâmetro $cross = FALSE$ sempre. Portanto, apenas os operadores **PathPlasmidio** e **TreePlasmidio** são utilizados. Na linha 8, as soluções de elite são adicionadas ao conjunto $Elite_i$. O procedimento **prepare_elite** seleciona 2 indivíduos de cada subpopulação e aplica o operador Transposon (busca local). Em seguida, a base $Host_{P,elite}$ é iniciada com os caminhos presentes nas árvores das soluções de elite na linha 9. Na linha 10, a melhor solução com base no custo é definida considerando todas as subpopulações (o * tem esse significado no algoritmo). Neste momento, o laço das linhas 11-29 é utilizado para criar as populações **offspring** conforme ilustrado nas Figuras 21 e 22. Para cada subpopulação $Subpop_{op}$, $|Subpop_{op}|$ novos indivíduos serão criados. O primeiro passo para criação de um indivíduo é escolher o método utilizado. Na linha 13, o *metodo* é definido aleatoriamente no intervalo (0,1). Se o valor obtido for menor que $Plasmid_{rate}$ um novo indivíduo será criado utilizando o plasmídio **PathPlasmidio**. Na linha 15, um indivíduo, T^{rand} , é escolhido aleatoriamente na subpopulação corrente. Na linha 16, uma base de informação é escolhida para ser utilizada pelo **PathPlasmidio**. Na linha 17, um indivíduo T^j é criado utilizando o Algoritmo **PathPlasmidio**. Caso contrário, Um indivíduo, T^{rand} é escolhido aleatoriamente na linha 23. Em seguida, o Algoritmo **TreePlasmidio** é utilizado na linha 24 para criar o indivíduo T^j . Por fim, o indivíduo T^j é adicionado a subpopulação $Subpop_{op}$ na linha 27. Este processo se repete para cada indivíduo de cada subpopulação na i -ésima iteração. Ao final da iteração, cada subpopulação (subpopulação + offspring) é unida às soluções de elite e em seguida filtradas utilizando o procedimento **fast_nondominated_sort()** nas linhas 31 e 32. Ao final da evolução das populações, o conjunto ND é definido na linha 36. Todas as subpopulações são incluídas em ND na linha 38. Por fim, o conjunto ND é filtrado utilizando o procedimento **fast_nondominated_sort** na linha 40.

Mota-Plasmid-Cross

Esta versão do algoritmo considera o uso do operador de cruzamento definido no Algoritmo 30. Além disso, o plasmídio **TreePlasmidio** não é utilizado. O fluxo de execução desta versão é exatamente o mesmo da versão anterior. Apenas com esta diferença.

A aplicação do operador de cruzamento necessita de três indivíduos selecionados aleatoriamente na linha 23. Em seguida, um novo indivíduo, T^j , é criado na linha 24. O restante dos processos é o mesmo que a versão anterior.

6.5 Resumo

No presente capítulo, foram apresentados os algoritmos propostos para os modelos estudados neste trabalho. Algoritmos para o problema de roteamento *multicast* otimizando

capacidade residual sujeito a limite de custo foram descritos. Assim como, uma heurística para o problema de roteamento *multicast* multissessão com múltiplas fontes. Por fim, foram apresentados operadores de cruzamento, mutação, plasmídio e um algoritmo transgenético para o problema de roteamento *multicast* multiobjetivo.

7 Resultados Experimentais

Neste capítulo, o processo de experimentação dos algoritmos nos modelos apresentados é descrito. Na Seção 7.1, apresenta-se o processo de criação de instâncias. Na Seção 7.2, é descrito o ambiente de experimentação utilizado no experimento. Na seção 7.3, ilustra-se o resultado obtido pelo *solver* Gurobi. Na Seção 7.4, é apresentado todo o processo de experimentação dos resultados para o modelo otimizando capacidade residual sujeito a custo. Na Seção 7.5, é apresentado o processo de experimentação no modelo proposto por Chen et al. (2013). Na seção 7.6, é apresentado o resultado dos algoritmos no modelo multiobjetivo.

7.1 Instâncias

Nesta seção, as instâncias utilizadas nas avaliações experimentais dos algoritmos propostos são descritas. Três conjuntos de instâncias foram definidos. Cada conjunto de instâncias está associado a um cenário de avaliação. Os cenários de avaliação estão associados aos modelos matemáticos estudados neste trabalho.

As instâncias foram geradas utilizando um processo em duas fases. Na primeira fase, a topologia foi gerada com base no modelo proposto por Waxman (1988). Este modelo permite construir as topologias da seguinte forma. Um plano é definido com lado de tamanho L . Em seguida, os vértices são distribuídos de modo aleatório e uniforme neste plano. Por fim, uma função de probabilidade determina a criação de uma aresta (i, j) de acordo com a Equação 7.1:

$$P(i, j) = z \cdot e^{-d/w \cdot L} \quad (7.1)$$

Os valores z e w estão no intervalo $(0, 1]$. Estes valores são utilizados para determinar a densidade de arestas longas (distância no plano) e a densidade de arestas no grafo, respectivamente. d representa a distância entre os vértices i e j no plano considerando a possibilidade de conectá-los.

A segunda fase da geração das instâncias consiste em adicionar as informações dos grupos *multicast*. As seguintes informações são adicionadas:

- identificador do grupo - valor de 0 até $|W| - 1$;
- fonte(s) do grupo - determina $S_k \subset V$;
- clientes do grupo - determina $D^k \subset V \setminus S_k$;

- requerimento de tráfego - determina o valor de t^k . Geralmente percentual da capacidade máxima das arestas.

O algoritmo utilizado sorteia os participantes dos conjuntos S_k e D^k de modo aleatório e uniforme. O mesmo é realizado para o valor de t^k .

A primeira fase do processo de geração de instância foi auxiliada pelo *Framework* BRITE (Boston university Representative Internet Topology gEnerator) (MEDINA et al., 2001). Ele incorpora o modelo de Waxman e permite configurar os parâmetros da Equação 7.1.

7.1.1 Conjunto de Teste - Mono-Fonte

O conjunto de instâncias definido nesta seção possui 120 unidades no total. Elas foram definidas para explorar os algoritmos nos parâmetros n e W . O número de participantes por grupo *multicast* foi definido de modo uniforme e aleatório. Os parâmetros utilizados no modelo de Waxman para gerar essas instâncias foram: $L = 1000$, $z = 0.2$ e $w = 0.25$.

As instâncias estão divididas em quatro grupos de 30 unidades. Essa divisão é baseada no número de vértices, $n \in \{30, 60, 120, 240\}$. Além disso, cada subconjunto de instâncias é subdividido em conjuntos classificados pelo número de grupos *multicast*, $W \in \{5, 10, 15, 20, 25\}$. O número de fontes definidas por grupo é $S_k = 1$ para todos os grupos em todas as instâncias. O número de participantes, D^k , é um percentual do número de vértices do grafo. O Valor de requerimento de tráfego foi definido de forma aleatória e uniforme sendo $1 \leq t^k \leq 20$. Na Tabela 6, um resumo dos parâmetros utilizados nas instâncias é apresentado.

Tabela 6 – Conjunto de Instâncias 1.

V	$ W $	$ D^k $	E	$b_{(i,j)}(min, max)$
B030	5,10,15,20,25	20 a 30 % $ V $	$2V$	45-100
B060	5,10,15,20,25	15 a 30 % $ V $	$2V$	45-100
B120	5,10,15,20,25	10 a 30 % $ V $	$2V$	45-100
B240	5,10,15,20,25	10 a 20 % $ V $	$2V$	45-100

Na Tabela 6, *B030* indica o conjunto com instâncias cujo valor de $n = 30$. Cinco subconjuntos dessas instâncias foram definidos onde cada conjunto é caracterizado pelo valor de W . Cada um desses subconjuntos possui seis instâncias. O número de vértices participantes por grupo *multicast*, D^k , é definido de modo aleatório e uniforme com base no percentual ilustrado. O valor de capacidade das arestas, $b_{(i,j)}$, é definido de forma aleatória e uniforme no intervalo ilustrado na tabela. O mesmo raciocínio é usado nos demais grupos de instâncias. Uma instância é identificada da seguinte forma: *B30_05_1*, onde 30 é número de vértices, 5 é o número de grupos e 1 é o identificador.

7.1.2 Conjunto de Teste - Multi-Fonte

Este conjunto contém instâncias propostas por Chen et al. (2013) e instâncias criadas pelo autor deste trabalho de acordo com os parâmetros indicados por Chen et al. (2013). As instâncias são utilizadas para avaliar os algoritmos em 4 situações relacionadas ao número de vértices, n , número de participantes por grupo, D^k , número de grupos, W , e o número de fontes por grupo, S_k . Os parâmetros utilizados no modelo de Waxman são: $L = 2000$, $z = 0.15$ e $w = 0.2$.

Tabela 7 – Conjunto de Instâncias 2.

Conjunto	t^k	n	$ S_k $	$ D^k $	$ W $	$b_{(i,j)}$
Bloco 1	1	200	1	10,20,30,40,50	30	W
		200	1	30	20,40,60,80,100	W
		40, 80, 120, 160, 200	1	30	40	W
Bloco 2	1	200	5	30	10, 20, 30, 40, 50	W
		200	5	10, 20, 30, 40, 50	40	W
		40, 80, 120, 160, 200	5	30	40	W
Bloco 3	1,3,9	200	5	10, 20, 30, 40, 50	30	60
		200	5	40	20, 40, 60, 80, 100	100
		2000	10, 20, 30, 40, 50	30	40	60
		40, 80, 120, 160, 200	5	40	30	60
Bloco 4	1	2000	10	100, 200, 300, 400, 500	30	W
		2000	10	50	100,200,300,400,500	W
		2000	10, 20, 30, 40, 50	50	50	W
		4000, 8000, 12000, 16000	10	100	100	W

Na Tabela 7, as configurações das instâncias utilizadas no experimento da heurística MMVD são ilustradas. Os blocos de instâncias **1**, **2** e **3** foram definidas por Chen et al. (2013). O **Bloco 4** foi definido pelo autor deste trabalho visando ampliação do teste de escalabilidade dos algoritmos testados considerando valores de n maiores que aqueles apresentados por Chen et al. (2013).

Para cada conjunto de configuração definido nos blocos da Tabela 7, 20 instâncias foram desenvolvidas. Por exemplo, no Bloco 1 a configuração $V = 40$, $S_k = 1$, $D^k = 30$ e $W = 40$ possui 20 instâncias com essas mesmas características. A mesma quantidade de instâncias aplica-se as demais configurações. Ao todo, 1400 instâncias foram definidas neste conjunto.

7.1.3 Conjunto de Teste - Multiobjetivo

A avaliação dos algoritmos multiobjetivo foi realizada com dois conjuntos de instâncias. O primeiro conjunto utilizado foi definido na Seção 7.1.1, especificamente os conjuntos $B030$ e $B060$. Além disso, um novo conjunto foi criado especificamente para esta avaliação.

As instâncias definidas nesta seção consideram a avaliação de três parâmetros: o tamanho da rede, n , o número de participantes, D^k , e o número de grupos, W . Os parâmetros utilizados no modelo de Waxman foram $L = 1000$, $z = 0.2$ e $w = 0.25$.

Tabela 8 – Conjunto de Instâncias 3.

n	D^k	W
20, 40, 60, 80, 100	5% a 10% de V	15
100	10, 20, 30, 40, 50	15
100	10	10, 20, 30, 40, 50

As configurações ilustradas na Tabela 8 foram utilizadas nos experimentos com os algoritmos multiobjetivo. A capacidade das arestas, $b_{(i,j)}$, foi definida no intervalo $[45, 100]$ de modo aleatório e uniforme. O valor de requerimento de tráfego, t^k , foi definido de modo aleatório e uniforme no intervalo $[1, 8]$. Para cada configuração presente na tabela, 20 instâncias foram desenvolvidas totalizando 300 unidades.

7.2 Plataforma de Execução

Os experimentos realizados para comparação dos algoritmos projetados e descritos no Capítulo 6 foram conduzidos na seguinte plataforma computacional:

- Sistema operacional Ubuntu versão 15.04;
- Processador Intel(R) Core(TM) i7 - 2600K CPU 3.40GHz;

- Compilador *gcc* versão 4.8.4-2;
- Linguagem de programação *c++* versão 11/14.

Além disso, a biblioteca de algoritmos em grafos *Boost* foi utilizada para inclusão de componentes para tratamento de expressão regular e *heap* de Fibonacci (POLUKHIN, 2013). Todas as execuções realizadas utilizam apenas uma *thread*.

7.3 Resultados *Solver* Gurobi

O *solver* Gurobi foi utilizado para obter o resultado ótimo para os modelos estudados neste trabalho. Este solver é voltado para indústria e possui licença para uso comercial (OPTIMIZATION, 2015). Entretanto, uma licença para uso acadêmico pode ser obtida em sua página na *web*.

Ele foi aplicado ao modelo que trata da otimização da capacidade residual sujeito a limite de custo e ao modelo multiobjetivo para obtenção da fronteira de Pareto ótima. Os modelos foram implementados utilizando a API (*Application Programming Interface*) do Gurobi para a linguagem *c++*.

7.3.1 Otimização de Capacidade Residual Sujeito a Custo

O modelo descrito na Seção 5.1, otimização de capacidade residual sujeito a custo, foi resolvido utilizando o *solver* Gurobi. O processo de resolução foi realizado em duas etapas. A partir deste momento o modelo será denominado de *Modelo_Z*.

Na primeira etapa, otimizou-se o modelo sem levar a restrição de limite de custo em consideração. Isso foi realizado para obter-se o valor máximo de capacidade residual e ter uma estimativa do custo de configuração de solução empregada. Em seguida, na segunda etapa, um valor limite para o custo de configuração da solução foi definido da seguinte forma: dado o custo obtido na primeira etapa, considera-se apenas 80% deste valor como parâmetro para o modelo levando-se em conta a restrição de limite de custo.

Além disso, na primeira etapa de resolução do modelo, sem levar em conta a restrição de limite de custo, é possível saber o valor máximo de Z . Portanto, um modelo alternativo foi utilizado para encontrar um menor valor de custo impondo que a solução deve ter o maior valor de Z . Este modelo foi denominado de *Modelo_{custo}*. Ele é uma modificação do *Modelo_Z* onde se coloca como função-objetivo a somatória do custo das arestas utilizadas na solução e fixa-se que o valor de capacidade das arestas deve ser maior ou igual a Z através de uma restrição.

Os resultados obtidos pelo *solver* Gurobi nos modelos *Modelo_Z* e *Modelo_{custo}* são ilustrados nas Tabelas 9 e 10. A primeira coluna da tabela identifica a instância do problema, as colunas Z , B e $T(s)$ representam os resultados de capacidade residual, limite

de custo aplicado (80% do valor original) e tempo de processamento em segundos, respectivamente, para o modelo *Modeloz*. Por fim, as duas últimas colunas, *Custo* e *T(s)*, representam o valor mínimo de custo quando se considera o valor máximo de *Z* e o tempo de processamento utilizado para obter este valor.

Tabela 9 – Resultado do Gurobi para $n = 30$ (esquerda) e $n = 60$ (direita).

Instância							Instância						
<i>Modeloz</i>			<i>Modelo_{custo}</i>				<i>Modeloz</i>			<i>Modelo_{custo}</i>			
B030	Z	B	T(s)	Custo	T(s)		B060	Z	B	T(s)	Custo	T(s)	
B30_05_1	3	16865	0,69	14617	0,19		B060_05_1	3	35555,2	2,12	26218	0,85	
B30_05_2	3	19530	0,37	16099	0,27		B060_05_2	4	23841,6	1,32	21890	0,80	
B30_05_3	3	18441	0,79	15767	0,29		B060_05_3	3	33460,8	23,63	26699	4,35	
B30_05_4	3	19895	0,50	17585	0,25		B060_05_4	3	30440,8	6,95	26171	1,11	
B30_05_5	3	20667	0,31	17495	0,28		B060_05_5	3	36996,8	2,22	28252	0,62	
B30_05_6	2	25062	0,66	20539	0,29		B060_05_6	3	33396	6,06	27240	1,88	
B30_10_7	7	36350	2,69	35481	0,56		B060_10_7	7	71819,2	55,87	63452	13,47	
B30_10_8	7	41886	1,82	39788	0,70		B060_10_8	7	79089,6	8,48	57156	12,19	
B30_10_9	7	43694	1,58	40437	0,65		B060_10_9	7	64348,8	11,58	53883	8,17	
B30_10_10	7	28954	1,56	27398	1,53		B060_10_10	7	62568,8	6,20	49303	2,03	
B30_10_11	7	35898	3,13	32710	3,62		B060_10_11	7	58506,4	5,45	46612	12,51	
B30_10_12	7	43867	4,04	37012	0,77		B060_10_12	7	68602,4	19,32	56541	2,61	
B30_15_13	11	62330	3,30	59396	1,71		B060_15_13	12	86847,2	130,89	80410	22,84	
B30_15_14	10	56207	3,05	55128	1,98		B060_15_14	11	105949,6	17,73	96800	8,79	
B30_15_15	11	58968	4,15	53537	5,24		B060_15_15	12	88721,6	136,62	87874	380,18	
B30_15_16	11	61487	11,96	60658	1,44		B060_15_16	12	99143,2	32,93	87805	40,30	
B30_15_17	11	66592	2,68	57719	2,04		B060_15_17	11	84184	12,96	68349	6,60	
B30_15_18	11	54286	6,94	52572	31,79		B060_15_18	11	90467,2	38,57	78371	9,39	
B30_20_19	14	91427	4,40	79707	1,97		B060_20_19	16	120351,2	156,77	109280	25,84	
B30_20_20	14	94090	4,56	81384	1,67		B060_20_20	15	139927,2	103,73	122293	23,09	
B30_20_21	15	82162	7,50	72754	1,48		B060_20_21	15	133903,2	85,99	121382	21,57	
B30_20_22	15	80450	8,98	79332	29,60		B060_20_22	16	114292	211,51	102655	71,96	
B30_20_23	16	65669	6,95	64798	1,26		B060_20_23	16	112691,2	89,60	102636	292,12	
B30_20_24	15	77758	9,56	69398	2,22		B060_20_24	15	127500	79,62	109884	24,09	
B30_25_25	18	85319	7,33	82341	10,15		B060_25_25	20	150852,8	137,98	132723	1521,79	
B30_25_26	19	82786	17,00	78232	15,91		B060_25_26	20	163195,2	156,38	154088	19020,21	
B30_25_27	18	109579	21,84	107069	10,25		B060_25_27	20	161916	95,24	142668	283,87	
B30_25_28	18	100157	10,33	98378	7,26		B060_25_28	20	167236,8	155,96	160767	237,74	
B30_25_29	19	95606	9,25	90053	2,30		B060_25_29	20	167788	253,19	155584	923,43	
B30_25_30	17	100130	8,43	85208	4,76		B060_25_30	20	145097	86,53	133260	635,34	

Vale salientar que o *Modelo_{custo}* foi otimizado utilizando um limite de tempo total de 3600s para as instâncias. Por este motivo, algumas instâncias possuem valor 0 no valor de custo da solução. Isso significa que o *solver* não foi capaz de encontrar uma solução inicial do problema.

7.4 Resultados do Roteamento Multissessão com Limite de Orçamento

Nesta seção, os resultados dos algoritmos descritos no Capítulo 6 e Seção 6.1 são apresentados. Todos os algoritmos projetos necessitam de afinamento de parâmetros para obtenção de seu melhor desempenho. Para tanto, foi utilizada uma ferramenta para este propósito. Na Seção 7.4.1, ilustra-se o processo de afinamento. Em seguida, na Seção 7.4.2, os resultados são apresentados.

Tabela 10 – Resultado do Gurobi para $n = 120$ (esquerda) e $n = 240$ (direita).

Instância						Instância					
		<i>Modelo_Z</i>		<i>Modelo_{custo}</i>				<i>Modelo_Z</i>		<i>Modelo_{custo}</i>	
B120	Z	B	T(s)	Custo	T(s)	B240	Z	B	T(s)	Custo	T(s)
B120_05_1	4	52755,2	372,73	46857	3605,64	B240_05_1	4	59293	566,12	51014	3600,98
B120_05_2	3	58486,4	23,20	45184	21,57	B240_05_2	4	64216	158,42	50323	3600,31
B120_05_3	3	53894,4	51,02	40712	24,16	B240_05_3	4	63716	229,21	53852	3600,27
B120_05_4	3	48218,4	26,57	35121	18,37	B240_05_4	3	69941	47,53	49561	1299,10
B120_05_5	3	63248,8	62,65	44050	1941,74	B240_05_5	4	66592	137,63	53698	3600,56
B120_05_6	4	40332,8	72,46	39758	2726,10	B240_05_6	4	67132	199,37	48198	3600,02
B120_10_7	8	109900	464,69	99604	3628,21	B240_10_7	8	131764	496,54	94049	3600,03
B120_10_8	7	96555,2	213,61	77170	49,82	B240_10_8	8	126718	687,79	98465	3600,05
B120_10_9	8	98248,8	256,55	87911	3600,54	B240_10_9	8	137765	1588,69	103401	3600,03
B120_10_10	8	105366,4	243,38	91094	3600,87	B240_10_10	8	137474	2811,19	99859	3600,01
B120_10_11	8	97852,8	681,64	79806	3600,38	B240_10_11	8	130686	3744,81	103681	3600,03
B120_10_12	7	99854,4	75,72	74018	2622,27	B240_10_12	7	141925	252,71	104018	581,66
B120_15_13	11	157984	222,93	130440	1703,74	B240_15_13	13	176598	1274,63	0	3600,05
B120_15_14	12	146200	1512,16	124314	3600,02	B240_15_14	13	190565	934,17	150440	3600,04
B120_15_15	12	156963,2	580,79	127519	3601,09	B240_15_15	13	185090	9881,42	157899	3600,04
B120_15_16	12	146739,2	1560,96	125335	3622,92	B240_15_16	13	182358	4249,81	157834	3600,05
B120_15_17	12	150414,4	1122,65	130078	3659,94	B240_15_17	13	186728	5102,05	149074	3600,02
B120_15_18	12	160987,2	1307,37	146448	3600,77	B240_15_18	13	175734	4876,45	156284	3600,03
B120_20_19	16	219908	400,65	181577	3600,88	B240_20_19	17	248086	1223,00	0	3600,04
B120_20_20	16	220070,4	1018,06	199090	3600,94	B240_20_20	17	230184	2223,48	179486	3600,05
B120_20_21	16	216052	2321,50	193370	3673,91	B240_20_21	17	252057	897,88	196456	3600,09
B120_20_22	16	210091,2	4614,09	185432	3600,03	B240_20_22	17	256098	3946,29	213208	3600,07
B120_20_23	16	217341,6	1386,53	179032	3631,51	B240_20_23	17	251014	1499,74	202117	3600,04
B120_20_24	17	196260,8	16516,32	195613	3600,01	B240_20_24	17	270336	1661,71	202258	3600,05
B120_25_25	20	277461,6	1541,86	233057	3601,21	B240_25_25	21	322512	2346,13	257711	3664,06
B120_25_26	21	240384	4970,35	227974	3600,02	B240_25_26	22	320206	11841,16	283006	3600,93
B120_25_27	20	258637,6	1064,78	0	3600,03	B240_25_27	20	298262	482,81	217373	3669,72
B120_25_28	21	262984,8	4613,80	245415	3630,79	B240_25_28	21	329876	2586,82	0	3600,05
B120_25_29	20	261462,4	492,88	218675	3601,16	B240_25_29	20	352243	746,38	0	3600,07
B120_25_30	21	238339,2	1977,19	222426	3600,02	B240_25_30	22	308052	65378,42	263341	3602,63

7.4.1 Definição dos Valores de Parâmetros

O processo de afinamento de parâmetros dos algoritmos constitui uma etapa importante do processo de avaliação. Através de um bom conjunto de parâmetros, o algoritmo pode obter melhores resultados. Portanto, é necessário encontrar tal combinação de parâmetros.

A escolha do conjunto de parâmetros para os algoritmos projetados foi realizada por meio da ferramenta IRACE (LÓPEZ-IBÁÑEZ et al., 2011), (CÁCERES; LÓPEZ-IBÁÑEZ; STÜTZLE, 2014). Esta ferramenta permite a utilização do *Iterated Racing*, procedimento que auxilia na definição dos parâmetros. O pacote está disponível nos repositórios da linguagem R.

O processo de escolha de parâmetros inicia pela definição dos intervalos de valores que podem ser aplicados a cada um deles. O IRACE toma estes valores como entrada junto com o binário do algoritmo e as instâncias de avaliação. Os valores escolhidos para avaliação e definição estão ilustrados na Tabela 11.

Na tabela 11, são ilustrados os valores utilizados para cada parâmetro de cada algoritmo proposto. Estes valores foram utilizados no IRACE, e ao final do processo de busca por parâmetros uma configuração foi escolhida para cada um dos algoritmos.

Os resultados obtidos pelo IRACE foram os conjuntos de configurações de parâmetros

Tabela 11 – Parâmetros de entrada do IRACE.

Configurações de Parâmetros						
<i>Algoritmo Colônia de Formigas</i>						
max_iter	α	β	l	h	θ	γ
[25, 50, 75, 100]	[0.0, 10.0]	[0.0, 10.0]	[0.0, 1.0]	[0.0, 1.0]	[0.0, 1.0]	[0.0, 1.0]
<i>Algoritmo GRASP</i>						
max_iter	μ	$heur$				
[20, 40, 60, 80, 100]	[0.1, 1.0]	[0.1, 1.0]				
<i>Algoritmo Genético</i>						
max_iter	$cross$	mut	pop	ϱ	l	
[25, 50, 75, 100]	[0.1, 1.0]	[0.1, 1.0]	[24, 32, 40, 48, 56, 64]	[0.1, 1.0]	[0.1, 1.0]	
<i>Algoritmo ILS</i>						
max_iter	ϑ	ϖ	upd			
[20, 40, 60, 80]	[0.1, 1.0]	[0.1, 1.0]	[1, 10]			

para cada um dos algoritmos avaliados. Na Tabela 12, ilustra-se resultado de cada um dos algoritmos listados na Tabela 11.

Tabela 12 – Configurações de parâmetros utilizadas nos algoritmos.

Algoritmo	Configuração
Colônia de Formigas	$max_iter = 75$, $\alpha = 0.8$, $\beta = 0.2$, $l = 0.9$, $h = 0.01$, $\theta = 0.05$, $\gamma = 0.3$
GRASP	$max_iter = 100$, $\mu = 0.16$, $heur = 0.14$
Algoritmo Genético	$max_iter = 25$, $cross = 0.5$, $mut = 0.1$, $pop = 24$, $\varrho = 0.13$, $l = 0.3$
Algoritmo ILS	$max_iter = 80$, $\vartheta = 0.28$, $\varpi = 0.33$, $upd = 2$

7.4.2 Metodologia

Os algoritmos propostos para o problema de roteamento *multicast* otimizando capacidade residual sujeito a limite de custo foram comparados entre si sobre o conjunto de instâncias 1, que foi definido na Seção 7.1 e Tabela 6.

A primeira avaliação considerou a variação no número de sessões, W . O valor de W utilizado é definido na Tabela 6, sendo $W = \{5, 10, 15, 20, 25\}$. Neste caso, o valor de W foi considerado associado ao valor de n . Cada valor de W tem um total de 6 instâncias em cada conjunto de instâncias por tamanho de n .

A segunda avaliação considerou o parâmetro associado ao tamanho da rede, n . O comportamento dos algoritmos com relação a variação do valor de n leva em conta os 4 conjuntos presentes na Tabela 6, sendo $n = \{30, 60, 120, 240\}$. Nesta avaliação, cada conjunto é constituído de 30 instâncias.

A metodologia de avaliação dos algoritmos nas instâncias mencionadas e agrupadas utiliza o teste de Friedman por grupos de instância, W e n . Além disso, o teste *PostHoc* de Nemenyi (NEMENYI, 1963) é aplicado em caso de diferença verificada no teste de Friedman (FRIEDMAN, 1937). Resumidamente, as seguintes etapas foram realizadas para comparação dos algoritmos:

- 50 execuções de cada algoritmo/instância;
- A mediana das execuções indica o resultado do algoritmo na instância;
- Instâncias agrupadas por parâmetro (W e n);
- Aplicação do teste de Friedman (instâncias são blocos e algoritmos tratamentos);
- Aplicação do *PostHoc* de Nemenyi quando se verificar diferença entre algoritmos.

Por exemplo, nas Figuras 26 e 27 e Tabela 29 no apêndice A, é apresentado o resultado da avaliação dos algoritmos diante da variação de W com $n = 30$. Nas Figuras 26 e 27, é ilustrado o valor de capacidade residual e tempo de processamento dos algoritmos. Para cada valor de W utilizado, 6 instâncias foram consideradas. Os algoritmos foram executados 50 vezes nestas instâncias e os gráficos ilustram a mediana obtida. O teste de Friedman toma como entrada as medianas de cada algoritmo em cada uma das instâncias. O resultado obtido indicou que há diferença e, portanto, o teste *PostHoc* de Nemenyi foi aplicado. Na Tabela 29, o resultado do teste de Nemenyi é ilustrado para todos os valores de W levando em conta os critérios de comparação: custo, tempo de processamento e capacidade residual. Todas as avaliações a seguir foram obtidas por meio do processo metodológico mencionado.

7.4.3 Avaliações

Número de Sessões

Todos os gráficos de tempo de processamento e capacidade residual, assim como os resultados do *PostHoc* de Nemenyi estão presentes no Apêndice A. Os resultados obtidos nesta avaliação indicam que os algoritmos ILS e o Genético tendem a ser melhores que o ACO. Em relação ao valor de capacidade, o ACO não foi superado pelo ILS nos seguintes casos $W = 5$ em todos os tamanhos de rede. Já em relação ao Genético, a diferença não foi verificada nos seguintes casos $W = 5$ com $n = 30$, $W \in \{5, 10, 25\}$ com $n = 60$, $W \in \{5, 10, 15\}$ com $n = 120$ e $W \in \{10, 20\}$ com $n = 120$, ilustrados nas Tabelas 29-32 respectivamente. Nas demais comparações entre os pares de algoritmos, não houve diferenças significativas.

O custo de configuração foi considerado na avaliação. Mesmo se tratando de um limite superior de custo a ser utilizado na configuração das soluções, este valor pode ser reduzido mantendo o valor de capacidade residual alto. Os resultados indicam que nos casos onde

$n \in \{30, 60\}$ a mudança no valor de W não afetou o custo de configuração obtido. Os algoritmos apresentam resultados equivalentes. A medida que n aumenta, o ILS e o Genético apresentam melhores resultados. O ILS consegue obter melhores resultados que o ACO e GRASP com $n = 120$ e $W \in \{5\}$. Além disso, o ILS obtém melhores resultados que o ACO com $n = 240$ e $W \in \{5, 15, 20\}$. Nos demais casos, notou-se equivalências conforme ilustra-se nas Tabelas 29-32.

Por fim, o tempo de processamento utilizado para encontrar as soluções foi comparado. O ILS foi equivalente ao ACO em praticamente todos os casos, exceto com $n = 240$ e $W \in \{15\}$. O ILS foi melhor que o GRASP com $n = 60$ em todos os casos, com $n = 120$ e $W \in \{5, 10\}$ e com $n = 240$ e $W \in \{5\}$. Nos demais casos, houve equivalências.

Diante dos resultados expostos no Apêndice A e o resumo realizado nos parágrafos anteriores, percebe-se que o algoritmo ILS foi, em geral, melhor que os demais algoritmos. Isso pode ser concluído considerando que ele obteve resultados pelo menos tão bom quanto os demais, sendo o valor de custo também competitivo. Outro fator importante em favor do ILS é o tempo de processamento. Neste critério o ILS foi melhor que GRASP e GENETICO, sendo equivalente ao ACO. No entanto, o ACO obteve valores de Z estatisticamente inferiores.

Número de Vértice

Os algoritmos foram comparados com relação a variação no tamanho da rede, onde $n \in \{30, 60, 120, 240\}$, 30 instâncias por valor de n . Esta avaliação foi realizada em conformidade com a metodologia definida no início da seção.

Os resultados obtidos foram comparados considerando a função-objetivo do problema, otimizar Z . Além disso, também considerou-se o custo de configuração e o tempo de processamento dos algoritmos. Na Tabela 33 do Apêndice A, os resultados do teste de *PostHoc* de Nemenyi são ilustrados para as métricas avaliadas.

A comparação realizada entre os algoritmos com relação ao valor de capacidade residual diante da variação do valor de n indica que os Algoritmos ILS e Genético sobressaem-se sobre o ACO e GRASP. O ILS e o Genético conseguem obter melhores resultados que o ACO em todos os casos. O ILS foi melhor que o GRASP em $n \in \{120, 240\}$ indicando que o aumento no número de vértices pode ampliar a vantagem do ILS. O Genético consegue superar o GRASP apenas em $n = 240$.

Com relação ao custo de configuração, o ILS consegue obter melhores resultados que o ACO e o GRASP em todos os casos avaliados. Houve equivalência entre o ILS e o Genético, assim como entre o GRASP e o ACO. O Algoritmo Genético conseguiu obter melhores resultados que o GRASP e o ACO quando $n \in \{120, 240\}$.

Por fim, a avaliação dos algoritmos em relação ao tempo de processamento indica que o algoritmo ILS sobressaiu-se em relação aos demais. O ACO obteve melhores resultados

com $n \in \{30, 120, 240\}$ e $n \in \{60\}$ em relação ao Genético e ao GRASP, respectivamente. O GRASP obteve melhores resultados que o Genético nos casos onde $n \in \{30, 120, 240\}$.

As avaliações dos algoritmos com relação ao tamanho da rede indicam que o algoritmo ILS consegue obter resultados significativos frente aos demais. Este resultado é obtido utilizando menos tempo de processamento que os demais procedimentos enfatizando sua eficácia.

Tabela 13 – Comparação de tempo de processamento *Solver*/Algoritmo proposto.

Conjunto	ACO	GRASP	GENÉTICO	ILS
B30	10,16	6,68	2,85	15,65
B60	44,81	30,76	12,77	103,4
B120	224,23	212,03	72,2	1010,01
B240	246,17	334,91	129,42	1724,04

Na Tabela 13, a vantagem do ILS fica mais clara quando se compara o tempo médio de execução por tamanho de rede em relação aos valores obtidos pelo *Solver*. Cada valor da tabela representa a média de tempo gasto no conjunto de instâncias. O ILS consegue ser 15, 103, 1010 e 1724 vezes mais rápido que o *Solver* nos conjuntos com n igual 30, 60, 120 e 240, respectivamente.

7.5 Resultados do Roteamento Multissessão com Múltiplas Fontes

Nesta seção, os resultados da heurística MMVD, descrita na Seção 6.2, são apresentados. Esta heurística foi projetada para resolução do problema proposto por Chen et al. (2013). O modelo proposto consiste em otimizar a capacidade residual considerando múltiplas fontes por grupo, conforme descrito na Seção 3.4.2.2. O Modelo será denominado a partir de agora como *Modelo_{chen}*.

7.5.1 Metodologia

A avaliação da heurística MMVD foi realizada utilizando o conjunto de instâncias 2, ilustrado na Tabela 7. Os conjuntos foram criados para avaliar o comportamento da heurística diante de diferentes configurações da rede caracterizadas pelo número de vértices, n , número de sessões (grupos), $r = |W|$, número de fontes, $s = |S_k|$ e número de participantes, $d = |D^k|$.

A heurística MMVD foi comparada a duas heurísticas presentes na literatura: MM-Forrests proposta por Chen et al. (2013) e as heurísticas P-MCF e E-MCF propostas por Guo et al. (2017). Este último trabalho é focado na otimização do custo levando em conta a possibilidade de usar mais de uma fonte na sessão. Entretanto, as heurísticas P-MCF

e E-MCF foram adaptadas para tratar múltiplas sessões e otimizar a capacidade residual da rede.

Três métricas foram utilizadas para avaliar os algoritmos: capacidade residual, que está presente na função-objetivo do problema; custo de configuração da solução e o tempo de processamento utilizado. As heurísticas foram executadas em 30 instâncias de cada configuração presente na Tabela 7. A mediana dos resultados obtidos por configuração são apresentados. Além disso, o teste de Friedman foi realizado sobre os resultados, assim como o teste *PostHoc* de Shaffer (SHAFFER, 1986). Resumidamente, o procedimento de coleta das amostras e aplicação dos testes seguiu os seguintes passos:

- Cada parâmetro possui 5 valores (considere $d = 10, 20, 30, 40, 50$);
- Para cada valor de d , 30 instâncias foram constituídas;
- Cada algoritmo rodou uma vez por instância;
- Os 30 resultados formam a amostra do algoritmo no parâmetro $d = 10$, por exemplo;
- A mediana destes valores é obtida;
- O teste de Friedman é aplicado nestes valores considerando os algoritmos como tratamentos e as instâncias como blocos;
- O teste de *PostHoc* de Shaffer é aplicado em caso de diferença apontada pelo Teste de Friedman.

Os resultados são exibidos com mais detalhes no Apêndice B. Na Tabela 34, no Apêndice B, as medianas dos algoritmos nas amostras coletadas para cada valor de r, d e n são apresentadas para a capacidade residual, tempo de processamento e custo. Além disso, o resultado estatístico é apresentado na Tabela 35, onde se compara o MMVD aos demais algoritmos com nível de significância igual a 0,05. Os demais resultados são apresentados do mesmo modo.

7.5.2 Cenário 1 - Multissessão Mono-Fonte: $t^k = 1, s = 1$

Neste cenário de avaliação, considerou-se o problema de roteamento *multicast* com múltiplas sessões e apenas uma fonte por sessão. Além disso, o requerimento de tráfego foi definido como $t^k = 1$.

A interpretação utilizada para avaliar os resultados é a seguinte. Considere-se o Bloco 1 definido na Tabela 7, onde há três conjuntos de configurações diferentes (cada linha da tabela representa um conjunto de configuração). Na primeira linha, tem-se a variação no valor de $d \in \{10, 20, 30, 40, 50\}$, onde $r = 30$ e $n = 200$. Para cada valor de d , 30 instâncias foram criadas. A mesma interpretação é dada para os demais parâmetros (n e

r) e também para os demais cenários (blocos de instâncias). Logo, os resultados ilustrados na Tabela 34 e 35, Apêndice B, representam as medianas das execuções das heurísticas em 30 instâncias para cada valor de d e o resultado estatístico obtido nesta amostra (tamanho 30), respectivamente.

A avaliação dos resultados com respeito ao valor de capacidade residual obtido pelas heurísticas diante da variação do valor de r neste cenário indica que não há diferença estatística entre o MMVD e o MMForests. Entretanto, percebe-se diferença para os dois outros algoritmos nos casos onde $r = 80$ e $r = 100$. Com relação ao custo de configuração, houve diferença significativa em relação ao P-MCF e E-MCF. O tempo de processamento obtido foi favorável ao MMVD, tendo em vista que houve diferença estatística em todos os casos comparados.

Resultado similar foi obtido diante da variação no número de participantes por grupo, d . O MMVD sobressaiu-se em relação ao P-MCF e E-MCF no valor de capacidade residual onde $d = 50$. Além disso, houve diferença entre o MMVD e o P-MCF e o E-MCF em todos os casos com relação ao custo e tempo de processamento. Já em relação ao MMForests, o resultado favoreceu o MMVD que foi melhor em todos os casos.

Por fim, a avaliação realizada diante da variação no número de vértices, n , indica que houve diferença entre o MMVD e as heurísticas E-MCF e P-MCF com relação ao valor de capacidade residual nos casos onde $n = 40$, e para todos os casos com relação ao custo e tempo de processamento. Na comparação com o MMForests nota-se que houve diferença nos casos onde $n \in \{40, 80, 120, 160\}$ em relação ao tempo de processamento.

7.5.3 Cenário 2 - Multissessão Multi-Fonte: $t^k = 1$, $s \geq 5$

Este cenário considera o uso de múltiplas fontes por grupo *multicast*. Cada grupo pode ser atendido por até 5 fontes. As instâncias do bloco 2 na Tabela 7 são utilizadas neste experimento. Os dados da avaliação estão resumidos nas Tabelas 36 e 37, Apêndice B, que apresentam as medianas dos resultados obtidos e a avaliação estatística conforme explicado na seção anterior.

Os resultados obtidos pelo MMVD ilustram a eficácia da heurística com relação a capacidade residual, custo e tempo de processamento diante da variação nos parâmetros r , d e n . Com relação ao valor de capacidade residual, apenas nos casos onde $r = 20$, $d \in \{10, 20, 40, 50\}$ e $n \in \{40, 80\}$ o MMVD não obteve resultados estatisticamente superiores as demais heurísticas. Por outro lado, há diferenças significativas entre o MMVD e as demais heurísticas com relação ao tempo de processamento e o custo de configuração em todos os casos avaliados.

Estes resultados indicam que o método de criação de soluções por meio da construção de diagramas de Voronoi apresenta-se como uma forma eficaz de resolver o problema proposto por Chen et al. (2013).

7.5.4 Cenário 3 - Multissessão Mono Fonte: $t^k \geq 1$, $s \geq 1$

Neste cenário de avaliação, foram consideradas múltiplas fontes por grupo *multicast* como ilustrado na seção anterior. Além disso, o requerimento de tráfego dos grupos é $t^k \geq 1$, especificamente o valor de $t^k \in \{1, 4, 9\}$. Esta avaliação utiliza as instâncias do bloco 3 na Tabela 7. A interpretação dos resultados é realizada da mesma forma que foi nos cenários 1 e 2.

Os resultados obtidos neste cenário são resumidos nas Tabelas 38 e 39, no Apêndice B, representam as medianas obtidas nas instâncias de cada configuração e os resultados do teste *PostHoc* de Shaffer, respectivamente. Os resultados avaliados dizem respeito a capacidade residual, custo e tempo de processamento das heurísticas.

Os testes indicam que o MMVD obteve resultados qualitativos conclusivamente superiores aos obtidos pela heurística MMForests com relação ao valor de capacidade residual nas instâncias dos parâmetros r, d, n e s em todos os casos. Além disso, ele mostrou-se tão eficiente quanto as heurísticas E-MCF e P-MCF no valor de capacidade residual obtido. Por outro lado, o MMVD produziu resultados significativamente melhores que as demais heurísticas em todos os casos avaliados com relação ao custo de configuração da solução e o tempo de processamento utilizado.

Estes resultados mostram que utilizar valores de $t^k > 1$ não tem impacto no desempenho da heurística MMVD em nenhuma das métricas consideradas na avaliação nem nos diferentes tipos de instâncias utilizadas.

7.5.5 Cenário 4 - Avaliação de Escalabilidade

Neste cenário, o propósito é avaliar as heurísticas com relação a escalabilidade. As instâncias do bloco 4 na Tabela 7 são de larga escala com relação ao número de participantes, d , número de vértices, n , número de grupos, r , e número de fontes por grupo, s . O valor de requerimento de tráfego é $t^k = 1$. A avaliação realizada neste cenário dá-se da mesma maneira que nos cenários anteriores.

As Tabelas 40 e 41, no Apêndice B, contém os resultados das heurísticas neste cenário. Na Tabela 40, é apresentada a mediana da execução dos algoritmos em cada conjunto de configuração definido. Na Tabela 41, o resultado do teste *PostHoc* de Shaffer é ilustrado.

A heurística MMVD saiu-se melhor que as demais heurísticas diante da variação no número de sessões, r , com relação ao valor de capacidade residual, custo e tempo de processamento.

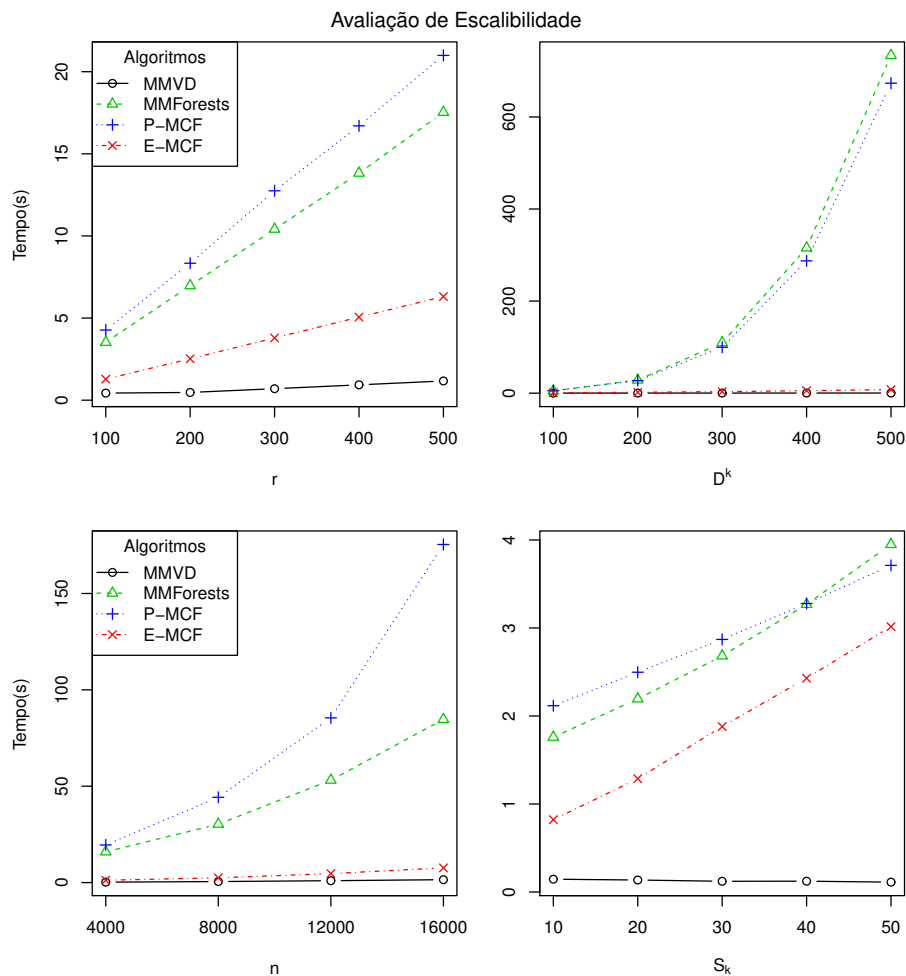
Na avaliação diante da variação no número de participantes, d , observou-se que o MMVD não obteve melhores valores que o E-MCF e P-MCF apenas quando $d = 400$ com relação ao valor de capacidade residual. Por outro lado, o MMVD obteve melhores resultados com relação ao custo e tempo de processamento que o E-MCF e o P-MCF em todos os casos. Já em relação ao MMForests, observa-se que o MMVD obteve melhores

resultados de capacidade residual nos casos onde $d \in \{300, 500\}$. Além disso, o custo de configuração e tempo de processamento obtidos pelo MMVD foram estatisticamente melhores que aqueles obtidos pelo MMForests.

A avaliação de escalabilidade diante da variação no valor de n indica que o MMVD foi superior as heurísticas E-MCF e P-MCF em todos os casos com relação a capacidade residual, custo de configuração e tempo de processamento. Já em relação ao MMForests, observou-se que não há diferença com relação aos valores de capacidade residual. Entretanto, o tempo de processamento é favorável ao MMVD em todos os casos e o custo de configuração dos algoritmos é também favorável quando $n \in \{4000, 8000, 12000\}$.

Na Figura 23, ilustra-se o tempo de processamento do MMVD em relação as demais heurísticas. Note-se que o MMVD consegue, em todos os parâmetros de configuração, obter um ótimo resultado com relação ao tempo de processamento, o que ocorreu nos demais cenários.

Figura 23 – Avaliação de escalabilidade com relação a tempo de processamento.



Fonte: própria.

Por fim, a avaliação de escalabilidade diante da variação no número de fontes, s , indica diferença apenas em $s \in \{40, 50\}$ em relação ao valor de capacidade residual. Já em relação ao custo de configuração e tempo de processamento há diferença estatística favorável ao MMVD em todos os casos.

7.6 Resultados do Roteamento Multissessão Multiobjetivo

Nesta seção, os resultados experimentais dos algoritmos propostos para solução do modelo multiobjetivo são apresentados. O modelo foi definido no Capítulo 5, Seção 5.2. A definição de parâmetros dos algoritmos é apresentada na Seção 7.6.1. Em seguida, apresenta-se a metodologia utilizada na Seção 7.6.3.1. Por fim, os resultados experimentais são apresentados nas Seções 7.6.3.2 e 7.6.4.2.

7.6.1 Definição de Valores de Parâmetros

O processo de afinamento de parâmetros foi realizado do mesmo modo que o processo descrito na Seção 7.4.1. Para cada algoritmo experimentado, um conjunto de parâmetros deve ser definido. A Tabela 14 contém os intervalos de valores dos parâmetros considerados para avaliação.

Tabela 14 – Valores utilizados como entrada para configuração de parâmetros pelo IRACE.

Algoritmo	N	iterações	HostPath	Plamid _{rate}	P _{rate}
Mota	[15, 30, 45, 60, 75, 90]	[50, 100, 150, 200, 250, 300]	[1,2,3,4,5,6]	[0,0 ; 1,0]	[0,0 ; 0,5]
gde	[10, 20, 30, 40, 50]	[50,100,150]			
moead	[100, 300, 500, 600]	[50, 100, 150, 200, 250, 300]			
sms	[100, 200, 300, 400, 500]	[50, 100, 150, 200, 250, 300]			
nsga2	[100, 200, 300, 400, 500]	[50, 100, 150, 200, 250, 300]			
ssnsga2	[100, 200, 300, 400, 500]	[50, 100, 150, 200, 250, 300]			

Na Tabela 15, o resultado obtido através do uso do IRACE é ilustrado. Estes são os valores de parâmetros utilizados pelos algoritmos nos experimentos realizados. Os parâmetros utilizados para as versões do MOTA são tamanho da população, número de iterações, *HostPath* que representa quantos caminhos serão armazenados nos bancos de informação, as duas taxa de aplicação de plasmídeos. Observe-se que o parâmetro **CROSS** e o número de soluções de elite utilizados são fixados em *TRUE/FALSE* e 6, respectivamente

Tabela 15 – Parâmetros de entrada dos algoritmos obtidos através do IRACE.

Algoritmo	N	iterações	HostPath	EliteSize	Plamid _{rate}	P _{rate}	cross
cross0	90	200	2	6	0,28	0,17	TRUE
cross1	90	150	1	6	0,52	0,29	TRUE
cross2	75	200	2	6	0,48	0,27	TRUE
cross3	75	250	1	6	0,33	0,41	TRUE
plasmid0	75	300	2	6	0,99	0,36	FALSE
plasmid1	60	300	1	6	0,9	0,45	FALSE
plasmid2	75	150	1	6	0,95	0,24	FALSE
plasmid3	60	300	2	6	0,9	0,7	FALSE
gde	50	150	-	-	-	-	-
moead	600	100	-	-	-	-	-
sms	100	300	-	-	-	-	-
nsga2	100	300	-	-	-	-	-
ssnsga2	100	300	-	-	-	-	-

7.6.2 Algoritmos

Na tabela 15, há 4 versões do algoritmo Mota que utilizam *crossover* e 4 versões que utilizam apenas plasmídio. Estas versões são caracterizadas pelos operadores de construção de árvores, $Op \in \{0, 1, 2, 3\}$. O valor 0 indica que haverá subpopulações de tamanho igual a $N/3$, ou seja, cada parte da população será constituída de soluções com árvores criadas com base nos operadores mencionados na Seção 6.3.2. As demais versões contêm apenas uma população caracterizada pelo tipo de operador de criação de árvores, neste caso tipos 1, 2 ou 3. O operador 1 define árvores utilizando caminho aumentado (maior capacidade possível nas arestas), o operador 2 define árvores com base no custo das arestas e o operador 3 define árvores com base no comprimento do caminho.

Os demais algoritmos utilizam os operadores de *crossover* e mutação (busca local). O algoritmo GDE utiliza o operador de *crossover* diferencial definido na Seção 6.3.3.5. Os demais algoritmos utilizam um operador de *crossover* que inclui os demais algoritmos de *crossover* presentes na Seção 6.3.3, neste caso cada operador de *crossover* tem a mesma probabilidade de ser utilizado (25% considerando que são 4 opções), tendo em vista que sorteia-se qual deles vai ser aplicado.

Na Tabela 16, os operadores utilizados pelos algoritmos são resumidos:

Os algoritmos foram avaliados em dois conjuntos de instâncias. O primeiro conjunto definido na Tabela 6 possui quatro tipos de redes classificadas pelo tamanho - $n \in \{30, 60, 120, 240\}$. Entretanto, apenas as instâncias com $n \in \{30, 60\}$ foram utilizadas. Estas instâncias foram resolvidas de modo a obter o conjunto de Pareto ótimo para cada uma delas. 30 instâncias foram utilizadas neste experimento.

O segundo conjunto definido na Tabela 8 foi utilizado integralmente e permite avaliar os algoritmos considerando a variação no número de participantes, D^k , número de vértices,

Tabela 16 – Algoritmos e operadores utilizados nos experimentos.

Algoritmo	Criação de Solução	Crossover	Crossover Diff	Mutação
cross0	1,2,3	-	X	Transposon
cross1	1	-	X	Transposon
cross2	2	-	X	Transposon
cross3	3	-	X	Transposon
plasmid0	1,2,3	-	-	Transposon
plasmid1	1	-	-	Transposon
plasmid2	2	-	-	Transposon
plasmid3	3	-	-	Transposon
gde	1,2,3	-	X	X
moead	1,2,3	X	-	X
sms	1,2,3	X	-	X
nsga2	1,2,3	X	-	X
ssnsga2	1,2,3	X	-	X

n , e número de sessões, W . Para cada valor de um dos parâmetros mencionados, foram criadas 20 instâncias. O total de instâncias utilizadas no experimento foi 300.

As métricas utilizadas para avaliação dos algoritmos foram I_H , I_ϵ e $I_{\epsilon+}$. Estes dados foram resumidos a partir dos conjuntos de aproximação gerados por cada algoritmo nas instâncias das Tabelas 6 e 8.

7.6.3 Experimento 1 - Avaliação de Instâncias gerais

7.6.3.1 Metodologia de Avaliação

O processo de criação e avaliação das amostras foi realizado da seguinte forma para as instâncias que não possuem o conjunto de Pareto ótimo definido:

- As instâncias são agrupadas por parâmetro (n , D^k e W);
- Cada valor de parâmetro de instância possui 20 unidades;
- Cada algoritmo foi executado 30 vezes em um instância;
- Os 30 conjuntos de aproximação foram transformados em um valor de qualidade por meio dos indicadores utilizados;
- Os 30 resultados obtidos na instância considerada foram resumidos por meio da mediana no indicador considerado;
- As 20 medianas são consideradas como entrada para comparação dos algoritmos no respectivo valor de parâmetro (por exemplo $n = 20$ com amostra de tamanho 20);

- Foi considerado o teste de Friedman e *PostHoc* de Shaffer sobre a amostra do item anterior com nível de significância igual a 0,05;
- O resultado final para cada conjunto de instâncias referente a um valor dos parâmetros n , D^k e W é resumido por meio de tabela de medianas e o mapa de calor com o teste de Shaffer.

Por exemplo, na Tabela 17, estão os resultados dos algoritmos em 20 instâncias da configuração com $n = 20$. Cada valor presente na tabela é a mediana do I_ϵ de 30 execuções na instância em questão. O valor 1,11 na linha 1 da coluna **cross0** indica a mediana do I_ϵ para as execuções naquela instância.

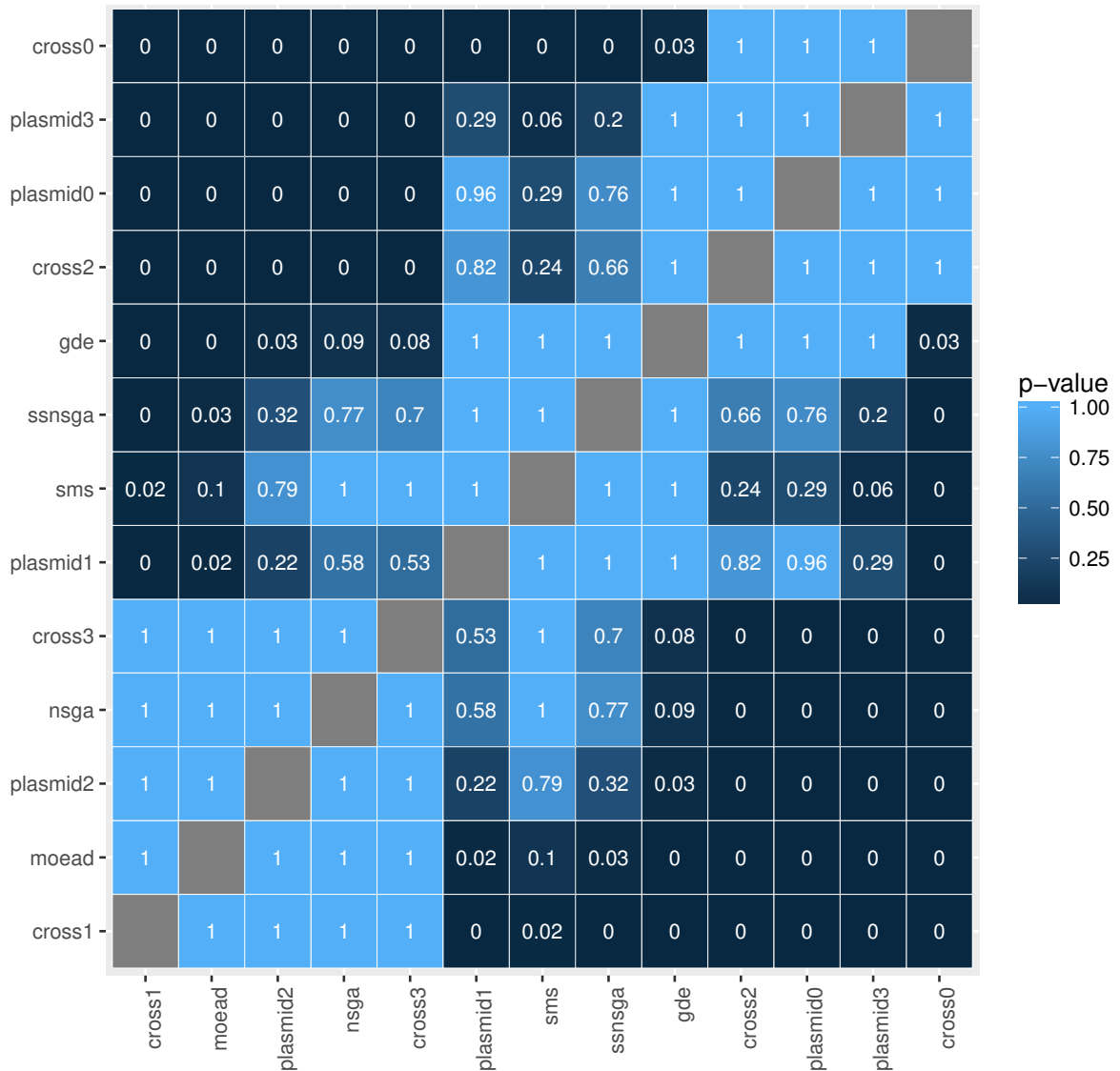
Tabela 17 – Mediana de 30 execuções da cada algoritmo nas instâncias com $n = 20$. Cada linha representa uma instância contendo a mediana dos algoritmos com relação ao valor do indicador I_ϵ .

Algoritmos													
n=20	cross0	cross1	cross2	cross3	gde	moad	nsga2	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.11	1.19	1.17	1.21	1.13	1.25	1.24	1.16	1.14	1.25	1.17	1.22	1.22
2	1.14	1.26	1.15	1.24	1.19	1.26	1.25	1.14	1.18	1.22	1.16	1.22	1.22
3	1.12	1.27	1.16	1.19	1.16	1.23	1.21	1.15	1.25	1.20	1.16	1.21	1.20
4	1.12	1.28	1.14	1.21	1.16	1.20	1.20	1.15	1.22	1.23	1.14	1.17	1.18
5	1.17	1.29	1.20	1.20	1.21	1.24	1.21	1.14	1.19	1.22	1.17	1.19	1.19
6	1.15	1.24	1.14	1.23	1.20	1.25	1.25	1.15	1.25	1.25	1.13	1.23	1.22
7	1.18	1.25	1.16	1.31	1.25	1.22	1.23	1.20	1.20	1.20	1.15	1.20	1.20
8	1.13	1.28	1.21	1.19	1.17	1.23	1.18	1.17	1.17	1.21	1.17	1.18	1.17
9	1.16	1.32	1.26	1.23	1.21	1.31	1.23	1.20	1.21	1.26	1.21	1.22	1.21
10	1.13	1.24	1.20	1.22	1.17	1.27	1.21	1.18	1.22	1.28	1.16	1.20	1.20
11	1.16	1.26	1.20	1.20	1.21	1.22	1.20	1.18	1.22	1.25	1.21	1.20	1.19
12	1.15	1.28	1.21	1.26	1.18	1.30	1.27	1.22	1.22	1.36	1.23	1.24	1.24
13	1.13	1.23	1.12	1.22	1.19	1.22	1.22	1.17	1.14	1.22	1.11	1.19	1.19
14	1.12	1.19	1.11	1.21	1.17	1.19	1.19	1.11	1.11	1.16	1.11	1.18	1.17
15	1.14	1.29	1.20	1.21	1.18	1.22	1.19	1.14	1.25	1.20	1.18	1.17	1.16
16	1.17	1.32	1.16	1.25	1.23	1.25	1.25	1.25	1.25	1.25	1.18	1.22	1.21
17	1.14	1.23	1.20	1.25	1.20	1.21	1.21	1.20	1.22	1.32	1.20	1.19	1.20
18	1.13	1.21	1.14	1.24	1.17	1.25	1.25	1.19	1.15	1.25	1.13	1.23	1.22
19	1.13	1.21	1.11	1.20	1.19	1.27	1.26	1.20	1.20	1.20	1.20	1.24	1.24
20	1.16	1.27	1.15	1.30	1.22	1.20	1.21	1.17	1.17	1.17	1.17	1.19	1.18

Os dados presentes na Tabela 17 ilustram as amostras dos algoritmos no conjunto com $n = 20$ com relação ao indicador de qualidade I_ϵ . Cada valor presente na tabela é a mediada de 30 execuções do algoritmo na respectiva instância.

As informações da Tabela 17 são utilizadas para realização do teste de Friedman. Neste exemplo, o resultado indicou que houve diferenças entre os algoritmos ao nível de significância 0,05. O resultado foi $p - value < 2.2e - 16$. Considerando que houve diferença, então o teste de *PostHoc* de Shaffer foi aplicado para verificar as diferenças entre pares de algoritmos na avaliação. Este resultado também é apresentado na Figura 24.

Figura 24 – Ilustração de resultados por meio de Mapa de calor.



Na Figura 24, o mapa de calor ilustrando o resultado do teste de Shaffer é apresentado. A comparação é realizada entre pares de algoritmos. O valor 0 indica um p-valor menor que 0.01. Em geral, o algoritmo **cross0** obteve melhores resultados. Apenas contra o **cross2**, **plasmid0** e **plasmid3** não houve diferença estatística com relação ao valor de I_ϵ .

Por fim, os resultados obtidos pelos algoritmos passam por processo de contagem para verificar em que situações cada um dos algoritmos foi melhor estatisticamente que os

demais. Na Tabela 18, apresenta-se a contagem de melhores resultados.

Tabela 18 – Contagem de resultados em relação ao indicador I_ϵ no tamanho da rede (n).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitórias(%)
cross0	0	4	4	5	1	5	5	0	5	5	4	5	5	80
cross1	0	0	1	0	0	4	1	0	0	3	0	0	0	15
cross2	0	1	0	1	0	2	1	0	0	1	0	0	0	10
cross3	0	0	0	0	0	2	0	0	0	2	0	0	0	6.7
gde	0	1	2	0	0	5	4	0	1	5	1	3	1	38.3
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid0	0	1	1	1	0	5	2	0	0	4	0	0	0	23.3
plasmid1	0	1	0	0	0	3	0	0	0	1	0	0	0	8.3
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	1	0	1	0	3	1	0	0	3	0	0	0	15
sms	0	1	0	0	0	0	0	0	0	0	0	0	0	1.7
ssnsga	0	1	0	0	0	1	0	0	0	1	0	0	0	5
Derrotas	0	18.3	13.3	13.3	1.7	50	23.3	0	10	41.7	8.3	13.3	10	

O resultado apresentado ilustra o número de vezes em que cada algoritmo foi melhor que os demais no parâmetro n (considera-se todos os valores do parâmetro). As vitórias de um algoritmo são observadas em cada linha. Já as derrotas são contabilizadas por coluna. Por exemplo, o algoritmo **cross0** possui 47 vitórias contra os demais algoritmos. Isso representa 80% de êxitos nas comparações. Já o número de derrotas pode ser observado na coluna. O **cross0** não foi inferior estatisticamente a nenhum dos algoritmos na variação do tamanho da rede e no indicador I_ϵ . A linha *Derrotas* e a coluna *Vitórias* representam valores percentuais. As Tabelas 102-113 contém todos os resultados de contagem para os parâmetros n , D^k e W nos indicadores de qualidade e tempo de processamento.

Os resultados obtidos pelos algoritmos serão ilustrados desta maneira para todos experimentos realizados nos conjuntos de instâncias mencionados no início da seção. Todos os resultados são apresentados no Apêndice C. No restante da seção, serão discutidos os resultados e apresentadas as conclusões.

7.6.3.2 Resultados

A avaliação dos resultados da comparação entre os algoritmos vai levar em conta os dados apresentados nas Tabelas 42-101, no Apêndice C. Eles representam as medianas dos indicadores de qualidade obtidos das amostras dos algoritmos. Além disso, serão utilizadas as informações do *Posthoc* de Shaffer ilustradas nas Figuras 34-93, no Apêndice C. O resultado de *Posthoc* de Shaffer será utilizado para contagem de vitórias e derrotas dos algoritmos. Se houver diferença estatística, o algoritmo com menor mediana obtém a

vitória, caso contrário contabiliza derrota. Esses dados estão resumidos por parâmetros da instância (n , D^k e W) nas Tabelas 102-113 no Apêndice C.

Os resultados obtidos e ilustrados nas Tabelas 102-113 indicam que versão **cross0** do algoritmo transgenético apresenta os melhores resultados nos indicadores de qualidade utilizados. Os resultados a seguir indicam o percentual obtido somando todas as avaliações favoráveis a um algoritmo (vitórias) - resultado agrupado por parâmetro sem considerar as subdivisões do conjunto de instâncias.

Com relação a avaliação no conjunto de instâncias classificadas por tamanho da rede, n , observa-se que o **cross0** obteve 78,3%, 80% e 78,3% de melhores resultados na comparação com os demais algoritmos nos indicadores de qualidade $I_{\epsilon+}$, I_{ϵ} e I_H , respectivamente. Outros dois resultados expressivos foram alcançados. A versão **plasmid0** obteve 53,3% de melhores resultados no I_H na comparação com outros algoritmos. O **gde** obteve 40% de melhores resultados no indicador $I_{\epsilon+}$.

Na avaliação dos algoritmos nas instâncias classificadas pelo número de sessões, W , os resultados foram em parte similares. A versão **cross0** obteve 61,7%, 61,7% e 65% de melhores resultados na comparação direta com os demais algoritmos nos indicadores $I_{\epsilon+}$, I_{ϵ} e I_H , respectivamente. Além disso, as versões **plasmid0** e **cross2** obtiveram 36,7% e 38,3% no indicador I_H , respectivamente. Por fim, destaca-se o **gde** com 33,3% no indicador $I_{\epsilon+}$.

Já na avaliação relacionada ao número de participantes por sessão, D^k . Observa-se, novamente, que a versão **cross0** apresenta os melhores resultados com 71%, 71,7% e 76,7% de melhores resultados nos indicadores $I_{\epsilon+}$, I_{ϵ} e I_H , respectivamente. Destaca-se também as versões **cross1** e **cross3** com 30% e 28,3% de melhores resultados no indicador de qualidade I_H , respectivamente. A versão **plasmid0** obteve novamente bom resultado com 40% de melhores resultados no indicador I_H .

Por outro lado, as versões do algoritmo transgenético demonstraram alto tempo de processamento. De modo geral, os algoritmos não obtiveram êxito neste quesito. Destaca-se, dos algoritmos já mencionados nos parágrafos anteriores, o **gde** que obteve 61,7%, 60% e 65% de melhores resultados no tempo de processamento nas instâncias dos conjuntos n , D^k , W , respectivamente.

Estas avaliações ilustram o comportamento dos algoritmos nos conjuntos n , D^k , W sem considerar a variação no valor do parâmetro. Nas Tabelas 19, 20 e 21, é ilustrado o número de vitórias e derrotas dos algoritmos por indicador de qualidade e subgrupos de instância dos conjuntos n , D^k , W . As colunas **v** e **d** ilustram o número de resultados positivos e negativos que o algoritmo obteve no conjunto de instâncias destacado. Por exemplo, a versão **cross0** obteve 8 vitórias e 0 derrotas no conjunto com $n = 20$ no indicador $I_{\epsilon+}$ conforme a Tabela 19. As colunas **V** e **D** apresentam a somatória de vitórias e derrotas, respectivamente.

Nas Tabelas 19-21, é possível observar como cada algoritmo comporta-se quando o

valor do parâmetro dos conjuntos de instâncias é alterado. Com relação ao parâmetro n , a versão **cross0** mostrou-se estável com relação ao número de vitórias diante do aumento de n . Isso se verifica para os indicadores $I_{\epsilon+}$, I_{ϵ} e I_H . O algoritmo **gde** mostra-se estável na avaliação referente ao $I_{\epsilon+}$. Nos demais casos, percebe-se que o algoritmo oscila na qualidade dos resultados quando n aumenta. O mesmo ocorre com o **plasmid0** que obtém resultados regulares no indicador I_H quando n aumenta. Nos demais indicadores mostra-se instável.

O comportamento dos algoritmos diante da variação no número de participantes, D^k , e no número de sessões, W , mostra que, de modo geral, o número de melhores resultados obtidos diminui gradativamente. Isso sugere que os algoritmos convergem para resultados mais similares conforme aumenta-se o valor de W e D^k , ou seja, a avaliação estatística não indica diferenças. Estes resultados são independentes do indicador de qualidade utilizado. Entretanto, a diferença de qualidade dos conjuntos de aproximação gerados ainda favorece a versão **cross0** do transgenético.

Todos os resultados obtidos indicam que a versão **cross0** é, sem dúvidas, uma solução potencialmente favorável de escolha dentre os demais algoritmos. Os resultados obtidos mostram que o algoritmo mantém certa regularidade na qualidade das soluções, independentemente do indicador de qualidade. Além disso, destaca-se também o uso de subpopulações caracterizadas por métodos de construção de solução diversificados. Isso é observável analisando as melhores versões **cross** e **plasmid** que foram **cross0** e **plasmid0**. Este último com resultado menos expressivo, mas sempre entre os algoritmos com as melhores avaliações.

Outro ponto importante da avaliação é a utilização do algoritmo de *crossover* diferencial. Apenas dois algoritmos utilizaram-no, o **gde** e o **cross0**. Os demais algoritmos utilizaram o mesmo *crossover* que permite aplicar diferentes operadores.

De todo modo, os resultados obtidos indicam que os elementos da transgenética favorecem à versão **cross0**. Primeiro, esta versão utiliza a ideia de subpopulações, que não está presente no **gde**. Segundo, os operadores de plasmídio e os bancos de informações permitem fazer manipulações não alcançáveis pelo **gde**. Terceiro, a diferença entre o **gde** e **plasmid0** não é tão ampla. No indicador de qualidade I_H , o **plasmid0** obteve melhores resultados em todos os conjuntos de instância. Por outro lado, o **gde** foi melhor nos indicadores $I_{\epsilon+}$ e I_{ϵ} em todos os conjuntos. Este resultado ressalta que os operadores transgenéticos têm contribuição para conjuntos de aproximação de qualidade.

Por fim, o tempo de processamento não foi favorável as versões do transgenético em geral. Um dos fatores é o uso do operador *transposon*. Este operador é utilizado em todas as iterações do algoritmo. Uma possibilidade de melhoria é considerar o uso de patamares para controlar o momento de aplicação do *transposon* e decidir qual o momento de aplicá-lo com mais frequência. Isso se justifica no fato de maior parte do tempo de processamento utilizado no transgenético ter sido destinado ao *transposon*.

Tabela 19 – Contagem de melhores resultados por algoritmo/indicador e número de vértices da rede (n).

I_{ϵ_+}												I_{ϵ}											

Tabela 20 – Contagem de melhores resultados por algoritmo/indicador no número de participantes por sessão (D^k).

Algoritmos	$I_{\epsilon+}$											
	$D^k = 10$		$D^k = 20$		$D^k = 30$		$D^k = 40$		$D^k = 50$		Soma	
	v	d	v	d	v	d	v	d	v	d	V	D
cross0	10	0	10	0	6	0	6	0	5	0	37	0
cross1	2	1	1	1	0	8	0	8	0	8	3	26
cross2	0	2	1	1	4	0	4	0	3	0	12	3
cross3	0	1	3	0	4	0	0	1	0	2	7	4
gde	6	0	3	0	4	0	4	0	3	0	20	0
moead	0	6	0	4	0	8	0	6	0	5	0	29
nsga	0	2	0	1	3	1	2	0	2	0	7	4
plasmid0	2	0	3	1	1	1	2	1	1	0	9	3
plasmid1	2	1	0	4	0	9	0	8	0	7	2	29
plasmid2	0	5	0	9	0	7	0	6	0	2	0	29
plasmid3	0	2	1	1	4	0	4	0	5	0	14	3
sms	0	2	1	1	4	0	4	0	2	0	11	3
ssnsga	1	1	1	1	4	0	4	0	3	0	13	2

Algoritmos	I_H											
	$D^k = 10$		$D^k = 20$		$D^k = 30$		$D^k = 40$		$D^k = 50$		Soma	
	v	d	v	d	v	d	v	d	v	d	V	D
cross0	10	0	10	0	8	0	6	0	5	0	39	0
cross1	2	2	1	2	0	5	0	4	0	8	3	21
cross2	2	2	3	1	7	0	6	0	5	0	23	3
cross3	2	1	3	1	4	0	1	0	3	0	13	2
gde	4	0	3	0	4	0	4	0	3	0	18	0
moead	0	9	0	5	0	5	0	3	0	4	0	26
nsga	0	3	1	2	1	2	1	2	1	2	4	11
plasmid0	8	0	6	0	4	0	2	0	2	0	22	0
plasmid1	2	1	0	5	0	9	0	9	0	10	2	34
plasmid2	0	8	0	9	0	5	0	3	1	2	1	27
plasmid3	0	3	0	2	1	2	1	2	2	0	4	9
sms	1	2	1	1	1	1	1	0	2	0	6	4
ssnsga	2	2	1	1	1	2	1	0	2	0	7	5

Algoritmos	I_{ϵ}											
	$D^k = 10$		$D^k = 20$		$D^k = 30$		$D^k = 40$		$D^k = 50$		Soma	
	v	d	v	d	v	d	v	d	v	d	V	D
cross0	10	0	11	0	6	0	5	0	5	0	37	0
cross1	0	1	0	1	0	9	0	8	0	8	0	27
cross2	0	2	0	1	4	0	3	0	2	0	9	3
cross3	0	2	3	1	2	1	0	2	0	2	5	8
gde	6	0	3	0	4	0	3	0	2	0	18	0
moead	0	5	0	3	0	7	0	2	0	3	0	20
nsga	0	2	1	1	4	0	3	0	2	0	10	3
plasmid0	2	0	2	1	2	1	2	0	2	0	10	2
plasmid1	2	1	0	6	0	9	0	8	0	8	2	32
plasmid2	0	5	0	8	0	7	0	7	0	2	0	29
plasmid3	0	2	1	1	4	0	3	0	5	0	13	3
sms	0	1	2	1	4	0	3	0	2	0	11	2
ssnsga	2	1	2	1	4	0	5	0	3	0	16	2

Algoritmos	Tempo de Processamento											
	$D^k = 10$		$D^k = 20$		$D^k = 30$		$D^k = 40$		$D^k = 50$		Soma	
	v	d	v	d	v	d	v	d	v	d	V	D
cross0	0	7	0	7	0	7	0	7	0	7	0	35
cross1	2	3	3	3	3	3	3	3	3	3	14	15
cross2	1	3	1	3	1	4	1	3	1	3	5	16
cross3	0	6	0	6	0	5	0	6	0	5	0	28
gde	8	0	7	0	7	0	7	0	7	0	36	0
moead	7	0	7	0	7	0	7	0	7	0	35	0
nsga	8	0	8	0	8	0	8	0	8	0	40	0
plasmid0	0	8	0	9	0	9	0	8	0	8	0	42
plasmid1	0	5	1	5	1	5	0	5	0	5	2	25
plasmid2	3	2	4	1	3	1	4	1	3	1	17	6
plasmid3	0	5	0	7	0	7	0	7	0	7	0	33
sms	5	0	5	0	6	0	5	0	5	0	26	0
ssnsga	5	0	5	0	5	0	5	0	5	0	25	0

Tabela 21 – Contagem de melhores resultados por algoritmo/indicador de qualidade no tamanho da sessão (W).

I_{ϵ_+}												I_{ϵ}																						
		W = 10		W = 20		W = 30		W = 40		W = 50		Soma				W = 10		W = 20		W = 30		W = 40		W = 50		Soma								
Algoritmos	v	d	v	d	v	d	v	d	v	d	V	D	Algoritmos	v	d	v	d	v	d	v	d	v	d	V	D	Algoritmos	v	d	v	d	v	d	V	D
cross0	9	0	10	0	9	0	7	0	8	0	43	0	cross0	9	0	10	0	9	0	6	0	9	0	43	0	cross0	9	0	10	0	9	0	43	0
cross1	6	0	1	1	1	1	4	0	5	0	17	2	cross1	4	0	0	1	0	1	3	0	5	0	12	2	cross1	4	0	0	1	0	1	12	2
cross2	0	4	1	1	1	1	0	8	0	9	2	23	cross2	0	4	0	1	1	1	0	9	0	9	1	24	cross2	0	4	0	1	1	1	9	1
cross3	0	1	2	1	4	0	4	0	4	0	14	2	cross3	0	3	1	1	2	0	4	0	3	1	10	5	cross3	0	3	1	1	2	0	10	5
gde	6	0	3	0	3	0	3	0	2	2	17	2	gde	7	0	3	0	2	0	4	0	2	2	18	2	gde	7	0	3	0	2	0	18	2
moead	0	5	0	9	0	9	0	9	0	6	0	38	moead	0	5	0	7	0	8	0	7	0	4	0	31	moead	0	5	0	7	0	8	0	31
nsga	0	4	0	3	0	3	2	1	3	1	5	12	nsga	0	3	0	2	0	1	2	1	3	1	5	8	nsga	0	3	0	2	0	1	5	8
plasmid0	6	0	3	0	1	1	3	1	4	1	17	3	plasmid0	6	0	2	0	1	1	3	0	3	1	15	2	plasmid0	6	0	2	0	1	1	15	2
plasmid1	2	1	2	1	1	0	4	0	4	0	13	2	plasmid1	2	1	2	1	1	0	4	0	4	0	13	2	plasmid1	2	1	2	1	1	0	13	2
plasmid2	0	5	0	5	0	3	0	9	0	9	0	31	plasmid2	0	5	0	4	0	3	0	9	0	9	0	30	plasmid2	0	5	0	4	0	3	0	30
plasmid3	0	4	1	1	1	1	0	4	0	8	2	18	plasmid3	0	4	1	1	1	1	0	5	0	8	2	19	plasmid3	0	4	1	1	1	1	8	19
sms	0	4	0	1	0	2	3	1	3	1	6	9	sms	0	2	0	1	0	1	2	1	3	1	5	6	sms	0	2	0	1	0	1	5	6
ssnsga	0	1	1	1	1	1	3	0	4	0	9	3	ssnsga	0	1	1	1	1	1	4	0	4	0	10	3	ssnsga	0	1	1	1	1	1	10	3

I_H												Tempo de Processamento																							
		W = 10		W = 20		W = 30		W = 40		W = 50		Soma				W = 10		W = 20		W = 30		W = 40		W = 50		Soma									
Algoritmos	v	d	v	d	v	d	v	d	v	d	V	D	Algoritmos	v	d	v	d	v	d	v	d	v	d	V	D	Algoritmos	v	d	v	d	v	d	V	D	
cross0	10	0	10	0	8	0	6	0	5	0	39	0	cross0	0	7	0	7	0	7	0	7	0	7	0	35	cross0	0	7	0	7	0	7	0	35	
cross1	2	2	1	2	0	5	0	4	0	8	3	21	cross1	3	3	3	3	3	3	3	3	3	3	15	15	cross1	3	3	3	3	3	3	15	15	
cross2	2	2	3	1	7	0	6	0	5	0	23	3	cross2	1	3	1	3	1	3	1	3	1	3	5	15	cross2	1	3	1	3	1	3	5	15	
cross3	2	1	3	1	4	0	1	0	3	0	13	2	cross3	0	7	0	6	0	6	0	6	0	6	0	31	cross3	0	7	0	6	0	6	0	31	
gde	4	0	3	0	4	0	4	0	3	0	18	0	gde	8	0	8	0	7	0	8	0	8	0	39	0	gde	8	0	8	0	7	0	39	0	
moead	0	9	0	5	0	5	0	3	0	4	0	26	moead	7	0	7	0	7	0	7	0	7	0	35	0	moead	7	0	7	0	7	0	35	0	
nsga	0	3	1	2	1	2	1	2	1	2	4	11	nsga	8	0	8	0	8	0	8	0	8	0	40	0	nsga	8	0	8	0	8	0	40	0	
plasmid0	8	0	6	0	4	0	2	0	2	0	22	0	plasmid0	0	8	0	8	0	8	0	8	0	8	0	40	0	plasmid0	0	8	0	8	0	8	0	40
plasmid1	2	1	0	5	0	9	0	9	0	10	2	34	plasmid1	0	5	0	5	0	5	0	5	0	5	0	25	plasmid1	0	5	0	5	0	5	0	25	
plasmid2	0	8	0	9	0	5	0	3	1	2	1	27	plasmid2	3	2	4	2	4	1	4	2	4	2	19	9	plasmid2	3	2	4	2	4	1	19	9	
plasmid3	0	3	0	2	1	2	1	2	2	0	4	9	plasmid3	0	5	0	7	0	7	0	7	0	7	0	33	plasmid3	0	5	0	7	0	7	0	33	
sms	1	2	1	1	1	1	1	0	2	0	6	4	sms	5	0	5	0	5	0	5	0	5	0	25	0	sms	5	0	5	0	5	0	25	0	
ssnsga	2	2	1	1	1	2	1	0	2	0	7	5	ssnsga	5	0	5	0	5	0	5	0	5	0	25	0	ssnsga	5	0	5	0	5	0	25	0	

7.6.4 Experimento 2 - Avaliação de instâncias com Pareto

7.6.4.1 Metodologia de Avaliação

O processo de avaliação dos algoritmos diante das instâncias com conjunto de Pareto definido foi realizado de modo similar ao experimento com instâncias gerais. Entretanto, o teste estatístico aplicado foi o teste de Kruskal (KRUSKAL; WALLIS, 1952). Em caso de diferença indicada no resultado, o teste de *Posthoc* aplicado foi o teste de Nemenyi (NEMENYI, 1963).

Esta avaliação considerou 30 instâncias de dois conjuntos definidos na Tabela 6. 15 instâncias foram selecionadas do conjunto com $n = 30$ e 15 do conjunto com $n = 60$. Estas instâncias são subdivididas pelo número de sessões W .

Na Tabela 22, as instâncias selecionadas para avaliação são detalhadas. Dois grupos de instâncias são formados pelo valor de n . A coluna ID indica o nome da instância e a coluna W indica o número de sessões. Além disso, tem-se o número de soluções encontradas pelo *Solver* Gurobi.

Tabela 22 – Instâncias com Pareto ótimo utilizadas no experimento de comparação dos algoritmos multiobjetivo.

n=30			n=60		
ID	Número de Soluções	W	ID	Número de Soluções	W
b30_01	43	5	b60_01	44	5
b30_02	57	5	b60_02	23	5
b30_04	34	5	b60_03	57	5
b30_07	71	10	b60_07	207	10
b30_08	127	10	b60_08	260	10
b30_09	106	10	b60_09	144	10
b30_13	200	15	b60_13	310	15
b30_14	73	15	b60_14	129	15
b30_15	172	15	b60_15	205	15
b30_19	217	20	b60_19	175	20
b30_20	150	20	b60_20	171	20
b30_21	66	20	b60_21	271	20
b30_25	146	25	b60_25	348	25
b30_26	116	25	b60_26	244	25
b30_27	92	25	b60_27	270	25

Os seguintes passos foram realizados:

- As instâncias são agrupadas por parâmetro (n, D^k) ;
- 15 instâncias para $n = 30$ e $n = 60$ foram escolhidas formando um conjunto de 30 instâncias.
- Este conjunto pode ser agrupado por valor de W . Quando isso é realizado 6 instâncias são utilizadas para cada valor de $W \in \{5, 10, 15, 20, 25\}$;
- Para cada instância/algoritmo 30 execuções são realizadas;

- O teste de Kruskal é aplicado para verificar se há diferença entre algum par de algoritmos nas amostras coletadas no passo anterior para a instância em questão;
- O teste *Posthoc* de Nemenyi é aplicado diante da existência de diferença;
- Em seguida, verifica-se o número de comparações em que um algoritmo obteve melhores resultados que os demais;
- Um resultado favorável a um algoritmo é verificado em duas etapas: se houver diferença estatística e a média da amostra na instância for menor que aquela do algoritmo comparado;
- O número de melhores resultados são computados por instância e nos parâmetros W e n .

Por exemplo, considere a Tabela 23 que indica as 30 execuções dos algoritmos na instância B30_01. O valor ilustrado é o I_H calculado considerando os conjuntos de aproximação gerados pelos algoritmos e o conjunto de Pareto da instância.

Os resultados ilustrados na Tabela 23 são a entrada utilizada no teste de Kruskal e no *Posthoc* de Nemenyi em caso de ser verificada alguma diferença entre qualquer par de algoritmos. O resultado obtido foi favorável a rejeição de hipótese nula, logo deve haver alguma diferença entre pelo menos um par de algoritmos. O resultado obtido foi $p.value = 2.6 \times 10^{-51}$.

Considerando que há diferença em pelo menos um par de algoritmos avaliado, o teste de *Posthoc* de Nemenyi é aplicado. O resultado é ilustrado no mapa de calor da Figura 25. Na figura, pode-se observar que a versão **cross0** obteve melhores resultados que 9 algoritmos. Apenas as versões **cross0**, **plasmidio0** e **plasmidio3** obtiveram os mesmos resultados e mostram-se estatisticamente equivalentes ao **cross0** na instância B30_01. Todos os resultados do experimento são detalhados no Apêndice D.

A partir destes resultados, é realizada a contagem de melhores resultados de cada algoritmo considerando os valores de parâmetros das instâncias, W e n , assim como considerando a avaliação por instância.

7.6.4.2 Resultados

Este experimento consiste em avaliar os algoritmos considerando instâncias cujo Pareto foi obtido. A avaliação considerou as métricas: I_H , I_ϵ , $I_{\epsilon+}$ e o tempo de processamento. Em geral, a versão do transgenético **cross0** obteve os melhores resultados considerando as instâncias classificadas por n e W . O resultado é detalhado no Apêndice D, nas Figuras 94-213 e Tabelas 116-155.

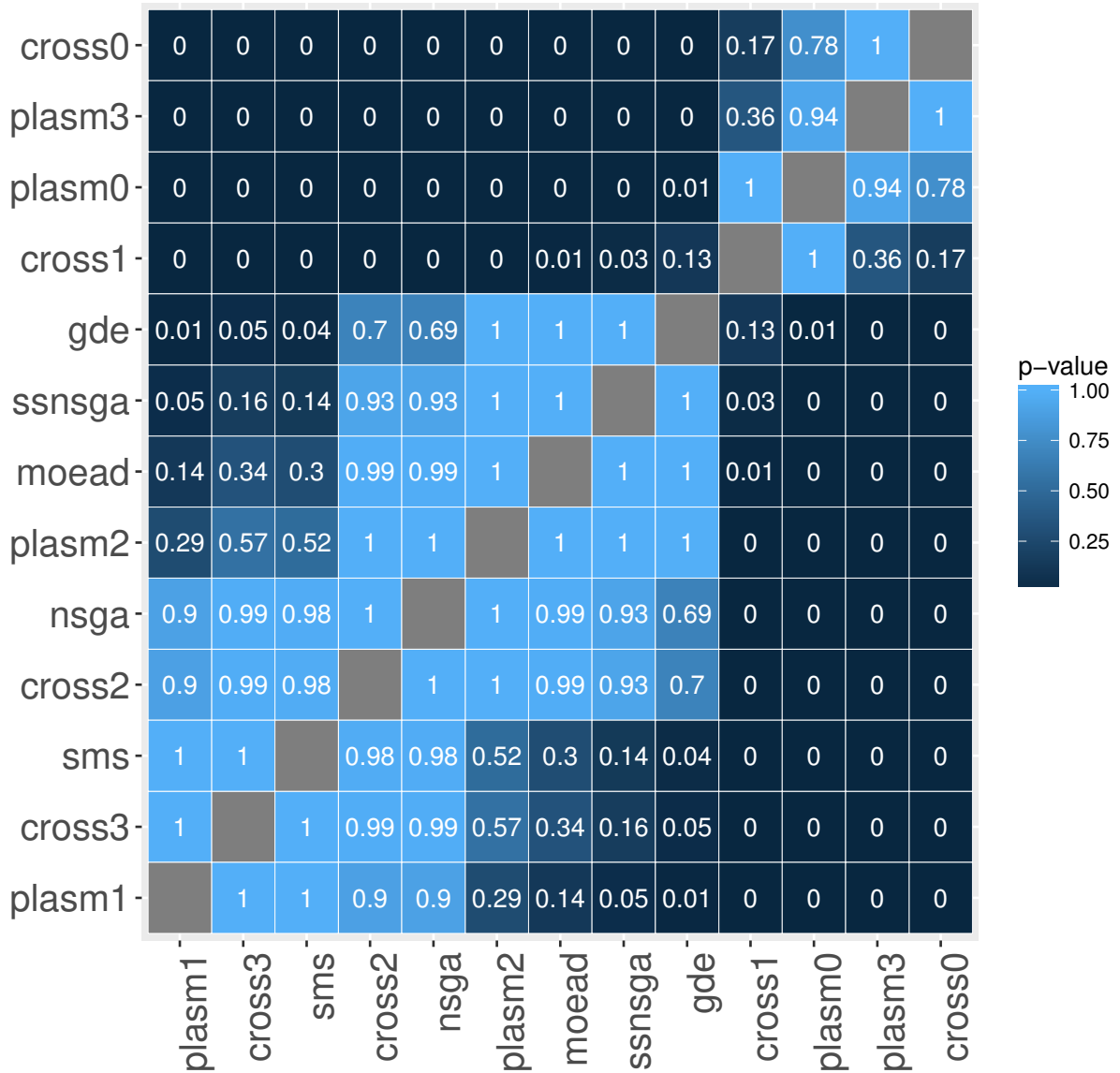
Nesta seção, um resumo dos dados analisados é apresentado levando em conta os resultados dos algoritmos diante da variação de W e também por tamanho de rede, n .

Tabela 23 – Hipervolume das execuções dos algoritmos na instância B30_01.

Execução	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moad	sms	nsga	ssnsga	gde
1	0.059	0.151	0.180	0.291	0.115	0.210	0.183	0.056	0.169	0.324	0.148	0.175	0.155
2	0.048	0.138	0.156	0.295	0.165	0.249	0.198	0.076	0.236	0.321	0.182	0.182	0.122
3	0.053	0.143	0.147	0.225	0.109	0.186	0.245	0.070	0.255	0.151	0.282	0.219	0.174
4	0.046	0.131	0.328	0.202	0.173	0.345	0.157	0.047	0.149	0.239	0.354	0.213	0.197
5	0.050	0.129	0.317	0.310	0.142	0.209	0.197	0.082	0.326	0.156	0.361	0.160	0.161
6	0.048	0.146	0.183	0.188	0.132	0.230	0.175	0.059	0.162	0.321	0.157	0.144	0.183
7	0.058	0.138	0.128	0.291	0.077	0.274	0.197	0.089	0.145	0.327	0.246	0.168	0.180
8	0.044	0.148	0.318	0.288	0.103	0.188	0.174	0.047	0.271	0.329	0.163	0.195	0.181
9	0.096	0.141	0.323	0.192	0.132	0.282	0.226	0.065	0.152	0.323	0.144	0.173	0.223
10	0.053	0.152	0.117	0.258	0.086	0.271	0.234	0.112	0.157	0.216	0.263	0.188	0.168
11	0.061	0.150	0.183	0.169	0.082	0.303	0.197	0.055	0.242	0.159	0.250	0.174	0.225
12	0.057	0.141	0.323	0.232	0.083	0.281	0.209	0.049	0.271	0.239	0.186	0.224	0.182
13	0.045	0.138	0.321	0.274	0.077	0.263	0.209	0.107	0.143	0.437	0.246	0.164	0.198
14	0.117	0.133	0.321	0.195	0.082	0.316	0.190	0.047	0.152	0.323	0.146	0.156	0.220
15	0.079	0.136	0.320	0.215	0.136	0.262	0.215	0.106	0.148	0.189	0.238	0.208	0.202
16	0.050	0.152	0.183	0.271	0.095	0.229	0.206	0.055	0.293	0.151	0.267	0.201	0.213
17	0.053	0.108	0.179	0.271	0.082	0.246	0.182	0.079	0.282	0.178	0.246	0.178	0.217
18	0.044	0.138	0.115	0.312	0.088	0.273	0.186	0.085	0.144	0.238	0.146	0.200	0.126
19	0.062	0.152	0.320	0.284	0.144	0.304	0.195	0.077	0.164	0.322	0.358	0.192	0.160
20	0.059	0.132	0.183	0.271	0.117	0.302	0.211	0.045	0.176	0.325	0.261	0.182	0.165
21	0.044	0.138	0.179	0.282	0.121	0.246	0.226	0.056	0.245	0.327	0.184	0.194	0.205
22	0.056	0.142	0.172	0.312	0.090	0.235	0.181	0.092	0.242	0.217	0.257	0.193	0.164
23	0.053	0.134	0.324	0.163	0.091	0.266	0.207	0.052	0.152	0.242	0.140	0.176	0.165
24	0.057	0.129	0.150	0.172	0.092	0.236	0.177	0.075	0.255	0.323	0.325	0.138	0.166
25	0.047	0.137	0.170	0.228	0.118	0.348	0.192	0.046	0.121	0.225	0.153	0.197	0.177
26	0.094	0.139	0.148	0.279	0.125	0.247	0.182	0.064	0.150	0.245	0.171	0.178	0.127
27	0.053	0.138	0.321	0.193	0.167	0.185	0.220	0.050	0.114	0.324	0.249	0.226	0.184
28	0.049	0.118	0.436	0.288	0.121	0.211	0.181	0.042	0.270	0.240	0.238	0.195	0.212
29	0.055	0.141	0.323	0.288	0.126	0.286	0.181	0.090	0.257	0.153	0.247	0.183	0.181
30	0.058	0.134	0.434	0.290	0.078	0.282	0.177	0.106	0.257	0.326	0.272	0.231	0.181
Média	0.058	0.138	0.244	0.251	0.112	0.259	0.197	0.069	0.203	0.263	0.229	0.187	0.180

Na Tabela 24, são apresentados os resultados dos algoritmos por grupo de instâncias considerando o valor de W nas métricas selecionadas para o experimento. A coluna $\mathbf{W} = 5$ é constituída de duas informações: número de melhores resultados dos algoritmos comparado aos demais nas instâncias com $\mathbf{W} = 5$. O máximo que pode ser obtido é 72. Isso se deve ao fato de haver 12 comparações para cada algoritmo em um total de 6 instâncias com $\mathbf{W} = 5$. A coluna % indica o percentual obtido considerando o valor máximo alcançável. Por exemplo, a versão **cross0** do transgenético obteve melhores resultados em 51 comparações o que representa 70,83% do máximo possível (se ela vencesse todas as comparações). O mesmo raciocínio é aplicado aos demais algoritmos.

Nas Tabelas 25-28, o resultado é resumido considerando os grupos de instâncias por tamanho de rede, n . Quatro tabelas são apresentadas, sendo uma para cada métrica utilizada. Por exemplo, na tabela referente ao indicador $I_{\epsilon+}$, a coluna $n = 30$ é subdividida

Figura 25 – Resultado do PostHoc de Nemenyi sobre os resultado da tabela 23 para o I_H na instância B30_01.

em duas colunas. A coluna V indica o número de vitórias do algoritmo naquele conjunto considerando todas as instâncias com $n = 30$. O número máximo que pode ser obtido é 180 vitórias considerando que são 12 comparações sobre um conjunto de 15 instâncias. Por exemplo, na tabela mencionada a versão **cross0** do algoritmo transgenético obteve 130 vitórias no conjunto com $n = 30$, o que representa um aproveitamento de 76,47%. A mesma interpretação é aplicada nos demais casos.

Os resultados obtidos e resumidos nas tabelas presentes nesta seção trazem fortes indícios de que a versão **cross0** apresentou vantagens significativas em relação aos demais algoritmos.

Considerando o indicador de qualidade $I_{\epsilon+}$ e a divisão das instâncias por valor de W , observa-se que o percentual de vitórias melhora a medida que W aumenta. Neste aspecto, registra-se 81,94% de aproveitamento nas comparações com os demais algoritmos

com $\mathbf{W} = \mathbf{25}$. Em geral, a média de aproveitamento obtida nas instâncias classificadas por W no indicador $I_{\epsilon+}$ foi de 73,89%. O segundo melhor resultado observado foi obtido pela versão **cross1** (construção de árvores com mais capacidade) com 57,22% de aproveitamento geral.

Em relação ao indicador I_{ϵ} , os resultados foram similares. A versão **cross0** obteve, novamente, resultados expressivos. Além disso, observa-se que o percentual de aproveitamento na comparação com os demais algoritmos melhora conforme aumenta-se o valor de W . O aproveitamento médio obtido pela versão **cross0** foi de 74,44%, levemente maior que na avaliação anterior. Novamente, a versão **cross1** obteve o segundo melhor aproveitamento com 54,72%. No entanto, o rendimento foi abaixo daquele obtido no $I_{\epsilon+}$.

Já em relação ao indicador I_H , os resultados obtidos pela versão **cross0** no aproveitamento geral foi consideravelmente melhor que nos casos anteriores. Observa-se que o percentual de aproveitamento quando $\mathbf{W} = \mathbf{25}$ foi de 84,72%. Na média geral, o **cross0** obteve um aproveitamento de 80,56%. As versões **cross1** e **plasmid0** obtiveram resultados expressivos entre os demais algoritmos com 51,39% e 47,22% na média geral, respectivamente.

Os resultados apresentados indicam que a versão **cross0** possui bom comportamento diante da variação do valor de W . O mesmo comportamento é observado se as instâncias forem agrupadas considerando o tamanho da rede, n . Essa informação é apresentada nas Tabelas 25-28. Para cada valor de n , é apresentado o total de vitórias obtidas por um algoritmo no conjunto considerando as 15 instâncias. O máximo de vitórias é 180 levando-se em conta as comparações com todos os algoritmos. O percentual de aproveitamento é ilustrado para cada algoritmo nas métricas consideradas. A versão **cross0** tem aproveitamento superior a 70% das comparações em todos os indicadores com $n = 30$ e $n = 60$, com destaque para o I_H quando $n = 60$. Este resultado ilustra que o desempenho do algoritmo é estável nos dois conjuntos conforme ilustrou-se previamente no resultado avaliado por W .

Por fim, a avaliação do tempo de processamento dos algoritmos indica que o comportamento nas instâncias com Pareto definido é o mesmo comportamento verificado nas instâncias gerais. No cenário atual, as melhores avaliações no tempo de processamento estão associadas aos algoritmos comparados com as versões do transgenético. Entretanto, observa-se que as versões **cross1** e **plasmid2** obtêm bons resultados.

As observações obtidas neste experimento consolidam as qualidades do algoritmo transgenético, especificamente a versão **cross0** que obteve resultados significativos nos dois experimentos: instâncias gerais e instâncias com Pareto. Esta versão obteve alta taxa de êxito nas comparações com os outros algoritmos do experimento.

Os resultados obtidos nas instâncias de Pareto são um forte indicativo da validade dos operadores de plasmídio projetados. Isso pode ser concluído com base nos dois experimentos. No primeiro experimento, o **gde** obteve excelentes resultados, justificando

a importância do *crossover* diferencial. No entanto, no segundo experimento apenas o **cross0** se manteve tão eficiente. Isso evidencia as qualidades dos operadores transgenéticos. Outra evidência em favor dos operadores transgenéticos é o fato de a versão **cross1** obter o segundo melhor aproveitamento, mostrando a eficácia do *crossover diferencial* aliado aos operadores transgenéticos. Além disso, a versão **plasmid0** também se mostrou eficiente no segundo experimento, inclusive obteve aproveitamento melhor que o **gde**. Esse resultado é importante pois ressalta outra característica presente apenas no **cross0** e **plasmid0**, que é o uso de subpopulações.

Tabela 24 – Resumo de contagem de vitórias e percentual total obtido por parâmetro W .

I_{ϵ_+}												I_{ϵ}													
		W = 10		W = 10		W = 15		W = 20		W = 25		Média	Algoritmos		W = 5		W = 10		W = 15		W = 20		W = 25		Média
		V	%	V	%	V	%	V	%	V	%	%	Algoritmos		V	%	V	%	V	%	V	%	V	%	%
cross0		51	70,83	48	66,67	52	72,22	56	77,78	59	81,94	73,89	cross0		51	70,83	47	65,28	52	72,22	58	80,56	60	83,33	74,44
cross1		29	40,28	50	69,44	44	61,11	46	63,89	37	51,39	57,22	cross1		26	36,11	51	70,83	42	58,33	45	62,50	33	45,83	54,72
cross2		26	36,11	7	9,72	15	20,83	14	19,44	17	23,61	21,94	cross2		25	34,72	7	9,72	15	20,83	13	18,06	17	23,61	21,39
cross3		14	19,44	13	18,06	24	33,33	23	31,94	26	36,11	27,78	cross3		14	19,44	13	18,06	22	30,56	23	31,94	26	36,11	27,22
plasmidio0		27	37,50	27	37,50	27	37,50	28	38,89	26	36,11	37,50	plasmidio0		26	36,11	25	34,72	26	36,11	29	40,28	29	40,28	37,50
plasmidio1		13	18,06	21	29,17	20	27,78	24	33,33	22	30,56	27,78	plasmidio1		13	18,06	20	27,78	21	29,17	24	33,33	22	30,56	27,78
plasmidio2		12	16,67	4	5,56	4	5,56	5	6,94	1	1,39	7,22	plasmidio2		11	15,28	4	5,56	3	4,17	5	6,94	1	1,39	6,67
plasmidio3		25	34,72	10	13,89	13	18,06	9	12,50	9	12,50	18,33	plasmidio3		26	36,11	10	13,89	13	18,06	6	8,33	8	11,11	17,50
gde		6	8,33	7	9,72	7	9,72	7	9,72	6	8,33	9,17	gde		6	8,33	7	9,72	7	9,72	5	6,94	5	6,94	8,33
sms		13	18,06	16	22,22	16	22,22	12	16,67	12	16,67	19,17	sms		14	19,44	16	22,22	18	25,00	12	16,67	12	16,67	20,00
ssnsga		10	13,89	14	19,44	13	18,06	9	12,50	12	16,67	16,11	ssnsga		11	15,28	14	19,44	13	18,06	11	15,28	10	13,89	16,39
nsga		13	18,06	17	23,61	14	19,44	13	18,06	14	19,44	19,72	nsga		11	15,28	18	25,00	14	19,44	15	20,83	13	18,06	19,72
moed		13	18,06	17	23,61	21	29,17	23	31,94	19	26,39	25,83	moed		13	18,06	16	22,22	21	29,17	22	30,56	20	27,78	25,56

I_H												Tempo(s)													
		W = 5		W = 10		W = 15		W = 20		W = 25		Média	Algoritmos		W = 5		W = 10		W = 15		W = 20		W = 25		Média
		V	%	V	%	V	%	V	%	V	%	%	Algoritmos		V	%	V	%	V	%	V	%	V	%	%
cross0		57	79,17	56	77,78	58	80,56	58	80,56	61	84,72	80,56	cross0		6	8,33	6	8,33	6	8,33	6	8,33	6	8,33	8,33
cross1		27	37,50	43	59,72	40	55,56	45	62,50	30	41,67	51,39	cross1		28	38,89	30	41,67	30	41,67	30	41,67	30	41,67	41,11
cross2		7	9,72	11	15,28	15	20,83	13	18,06	11	15,28	15,83	cross2		23	31,94	24	33,33	25	34,72	25	34,72	24	33,33	33,61
cross3		1	1,39	15	20,83	18	25,00	22	30,56	9	12,50	18,06	cross3		6	8,33	6	8,33	6	8,33	6	8,33	6	8,33	8,33
plasmidio0		35	48,61	34	47,22	35	48,61	31	43,06	35	48,61	47,22	plasmidio0		0	0,00	0	0,00	0	0,00	0	0,00	0	0,00	0,00
plasmidio1		12	16,67	19	26,39	20	27,78	21	29,17	19	26,39	25,28	plasmidio1		17	23,61	15	20,83	17	23,61	15	20,83	15	20,83	21,94
plasmidio2		3	4,17	5	6,94	1	1,39	4	5,56	0	0,00	3,61	plasmidio2		39	54,17	30	41,67	30	41,67	30	41,67	30	41,67	44,17
plasmidio3		24	33,33	18	25,00	13	18,06	7	9,72	8	11,11	19,44	plasmidio3		9	12,50	9	12,50	7	9,72	9	12,50	9	12,50	11,94
gde		9	12,50	9	12,50	7	9,72	6	8,33	11	15,28	11,67	gde		42	58,33	39	54,17	39	54,17	34	47,22	35	48,61	52,50
sms		25	34,72	14	19,44	17	23,61	17	23,61	26	36,11	27,50	sms		26	36,11	31	43,06	31	43,06	30	41,67	30	41,67	41,11
ssnsga		14	19,44	10	13,89	12	16,67	10	13,89	15	20,83	16,94	ssnsga		40	55,56	40	55,56	38	52,78	38	52,78	40	55,56	54,44
nsga		18	25,00	16	22,22	12	16,67	16	22,22	19	26,39	22,50	nsga		23	31,94	30	41,67	30	41,67	30	41,67	28	38,89	39,17
moed		21	29,17	18	25,00	22	30,56	25	34,72	23	31,94	30,28	moed		26	36,11	31	43,06	31	43,06	33	45,83	30	41,67	41,94

Tabela 25 – Contagem de melhores resultado no I_{ϵ_+} .

Algoritmos	I_{ϵ_+}			
	n = 30		n = 60	
	V	%	V	%
cross0	130	72.2	136	75.6
cross1	106	58.9	100	55.6
cross2	35	19.4	44	24.4
cross3	31	17.2	69	38.3
plasm0	72	40	63	35
plasm1	48	26.7	52	28.9
plasm2	15	8.3	11	6.1
plasm3	29	16.1	37	20.6
moead	16	8.9	17	9.4
sms	35	19.4	34	18.9
nsga	34	18.9	24	13.3
ssnsga	42	23.3	29	16.1
gde	56	31.1	37	20.6

Tabela 27 – Contagem de melhores resultado no I_H .

Algoritmos	I_H			
	n = 30		n = 60	
	V	%	V	%
cross0	139	77.2	151	83.9
cross1	100	55.6	85	47.2
cross2	35	19.4	22	12.2
cross3	38	21.1	27	15
plasm0	89	49.4	81	45
plasm1	49	27.2	42	23.3
plasm2	13	7.2	0	0
plasm3	36	20	34	18.9
moead	17	9.4	25	13.9
sms	38	21.1	61	33.9
nsga	28	15.6	33	18.3
ssnsga	38	21.1	43	23.9
gde	57	31.7	52	28.9

Tabela 26 – Contagem de melhores resultado no I_{ϵ} .

Algoritmos	I_{ϵ}			
	n = 30		n = 60	
	V	%	V	%
cross0	131	72.8	137	76.1
cross1	101	56.1	96	53.3
cross2	34	18.9	43	23.9
cross3	32	17.8	66	36.7
plasm0	73	40.6	62	34.4
plasm1	48	26.7	52	28.9
plasm2	13	7.2	11	6.1
plasm3	28	15.6	35	19.4
moead	15	8.3	15	8.3
sms	36	20	36	20
nsga	37	20.6	22	12.2
ssnsga	43	23.9	28	15.6
gde	55	30.6	37	20.6

Tabela 28 – Contagem de melhores resultado no Tempo.

Algoritmos	Tempo(s)			
	n = 30		n = 60	
	V	%	V	%
cross0	15	8.3	15	8.3
cross1	74	41.1	74	41.1
cross2	61	33.9	60	33.3
cross3	15	8.3	15	8.3
plasm0	0	0	0	0
plasm1	37	20.6	42	23.3
plasm2	80	44.4	79	43.9
plasm3	24	13.3	19	10.6
moead	100	55.6	89	49.4
sms	74	41.1	74	41.1
nsga	98	54.4	98	54.4
ssnsga	71	39.4	70	38.9
gde	72	40	79	43.9

7.7 Resumo

Neste capítulo, todos os detalhes de avaliação experimental foram apresentados. Ilustrou-se o processo de definição de instâncias, afinamento dos algoritmos e metodologia de avaliação. Por fim, foi ilustrado o resultado do processo experimental através de compilações dos resultados gerais ilustrados em apêndice.

8 Considerações Finais

Neste trabalho, foi realizado um estudo sobre o problema de roteamento *multicast* multi-sessão com múltiplas fontes. O estudo inclui dois novos modelos para o problema: um modelo mono-objetivo e um modelo multiobjetivo. Além disso, um terceiro modelo da literatura foi estudado. Em complemento ao estudo e criação de uma nova formulação matemática, vários algoritmos foram projetados para solucionar as novas versões do problema em questão.

O estudo realizado permitiu fazer um levantamento sobre o problema de roteamento *multicast* desde suas aplicações iniciais mais relevantes até as mais recentes contribuições presentes na literatura. Diversos modelos foram propostos para atendimento de demandas específicas de qualidade de serviço. Tais modelos empregam diferentes funções-objetivo e restrições. A partir deste levantamento, foram concebidos dois modelos matemáticos para o problema estudado.

O primeiro modelo visa a otimização da capacidade residual sujeito a um limite de custo – denominado de orçamento. O modelo é denominado *Modelo_Z*. Esta ideia surgiu da observação a respeito dos objetivos de otimização do problema que indicou, quase que exclusivamente, serem ou o custo de configuração da solução, ou uma função associada a conservação de recursos da rede. Este fato chamou a atenção pela predominância dessa divisão no tocante a definição da função-objetivo. Portanto, decidiu-se criar um modelo que permitisse explorar soluções envolvendo a capacidade residual e o custo de configuração simultaneamente. A solução escolhida foi definir a função-objetivo como a maximização da capacidade residual ao passo que se limita o custo de configuração da solução a um valor de orçamento.

O segundo modelo consiste de uma abordagem multiobjetivo para o problema estudado. Três funções-objetivo foram definidas considerando o valor de custo de configuração da solução, a capacidade residual da rede e a contagem de *hops* entre fonte e os clientes associados a cada árvore de distribuição. Este modelo tem como principal característica contemplar interesses de otimização que formam um importante trade-off, que são o custo e a capacidade residual. Adicionalmente, a contagem de *hops* é incluída como função-objetivo visando ao controle do crescimento dos caminhos entre fonte e destino na árvore de distribuição. Este objetivo é importante se for levado em consideração aplicações sensíveis ao *delay*. Além disso, a obtenção das soluções de Pareto do modelo proposto contempla soluções de modelos da literatura como é o caso dos modelos propostos por: Lee e Cho (2004), Kang, Park e Park (2009), Wang, Liang e Jan (2002) e Chen et al. (2013). Soluções destes modelos estão incluídas no Pareto. Isso torna o modelo proposto mais completo no sentido de prover uma forma de modelagem permitindo atingir diferentes interesses de otimização.

Além da modelagem de novas abordagens para o problema de roteamento *multicast*, algoritmos meta-heurísticos foram projetados para lidar com os novos modelos. Os algoritmos projetados formam três conjuntos distintos. O primeiro conjunto de algoritmos contempla aqueles projetados para modelo mono-objetivo, que tem como função-objetivo a maximização da capacidade residual. O segundo conjunto de algoritmos consiste de um algoritmo heurístico baseado em diagramas de Voronoi para o modelo proposto por Chen et al. (2013). Por fim, o terceiro conjunto de algoritmos inclui propostas de soluções para o problema multiobjetivo. Vários algoritmos foram projetados com base em arquiteturas de otimização multiobjetivo clássicas e também baseados na abordagem Transgenética.

Foram projetados quatro algoritmos para o problema de roteamento *multicast* na versão mono-objetivo: ACO, GRASP, Genético e ILS. Além disso, vários operadores de construção de solução, operador de mutação, operador de *crossover* e operador de busca local foram projetados. Eles foram avaliados através de uma abordagem experimental consistente que utilizou testes estatísticos para comparar as amostras geradas por cada algoritmo. Os resultados foram obtidos em um conjunto com 120 instâncias caracterizadas pelo tamanho da rede medido em número de vértices, $n = \{30, 60, 120, 240\}$ e pelo número de sessões, $W = \{5, 10, 15, 20, 25\}$. Os resultados obtidos indicam que o ILS e Genético possuem os melhores resultados com respeito ao valor de capacidade residual, assim como o valor de custo de configuração da solução. No entanto, o ILS possui tempo de processamento melhor em relação ao Genético e, portanto, pode ser mais indicado para utilização.

Os algoritmos projetados para o modelo multiobjetivo foram baseados nas arquiteturas clássicas NSGA2, MOEA/D, SMS-EMOA, GDE3. Além disso, foi utilizada a versão de estado estacionário do NSGA2, denominada ssNSGA2. Estes algoritmos foram utilizados em suas versões originais com as devidas modificações para adaptação ao problema de otimização tratado. Operadores necessários a essas arquiteturas foram projetados, tais como: operadores de criação de solução, operadores de *crossover* e busca local. Além dessas contribuições, um novo algoritmo transgenético foi projetado.

O algoritmo transgenético proposto possui duas variações definidas pelo uso de *crossover* ou sem uso de *crossover*, **cross** e **plasmid** respectivamente. Além disso, ele permite ser executado considerando quatro formas de montar a população inicial. Isso se deve ao fato de haver três operadores de construção de solução direcionados para cada um dos objetivos de otimização. Logo, a versão **plasmid** e a versão **cross** podem ter populações baseadas em um dos três objetivos. Esta população é unicamente formada por soluções associadas ao operador escolhido. Adicionalmente, foi projetado em ambas as versões a utilização de subpopulações independentes. Neste caso, uma população de tamanho N é dividida por 3 e cada parte da população é criada por um operador associado a um objetivo específico. Essas populações evoluem de forma independente. Quatro bases de informações genéticas foram criadas para guardar boas informações durante o processo

de evolução. Dois tipos de informação são guardados: caminhos e árvores. Elas são manipuladas via operadores de plasmídeo. Os caminhos podem estar associados ao custo ou a contagem de *hops*, ambos os menores possíveis. Uma solução é mantida durante a busca para guardar as árvores de menor custo já encontradas. Outro conceito empregado foi o de soluções de elite. Estas soluções são escolhidas por iteração e sofrem alterações via operador *transposon*. Em seguida, seu material genético é incluído nas bases de informação genética.

Os algoritmos multiobjetivo foram avaliados através de um extenso processo experimental. Dois conjuntos de instâncias foram utilizados para averiguar como os algoritmos se comportam diante de diferentes configurações da topologia da rede e também de informações *multicast*. Utilizou-se o hipervolume, o épsilon aditivo e multiplicativo e o tempo de processamento para avaliar os algoritmos. O primeiro conjunto de instâncias utilizado possui 300 unidades. Elas estão categorizadas por parâmetros: número de vértices, n ; número de sessões, W , e número de participantes em grupos, D^k . Cada um dos parâmetros possui 5 valores definidos. Estes experimentos permitiram verificar o desempenho dos algoritmos projetados a fim de observar os pontos fortes e o que precisa ser melhorado. Em geral, a versão **cross0** do algoritmo transgenético proposto obteve melhores resultados em praticamente todas as situações. Ela vem acompanhada do **plasmid0** e **gde** como algoritmos com bons resultados. Embora o **plasmid0** e **gde** tenham obtido resultados significativamente inferiores. O segundo conjunto de instâncias é formado por 30 unidades e categorizado pelo valor de n e W . Este conjunto teve o Pareto obtido para cada instância. O resultado obtido nesta versão coloca o **cross0** e **plasmid0** com algoritmos que se sobressaíram sobre os demais. Novamente, o **cross0** obteve os melhores resultados em todos os indicadores de qualidade e em, praticamente, todos os casos de teste. Por outro lado, observou-se que o algoritmo transgenético consumiu muito tempo de processamento, possivelmente devido ao uso do operador *transposon*.

Os resultados obtidos no experimento do modelo multiobjetivo indicam que os operadores da transgenética têm grande contribuição para os ótimos resultados alcançados pelas versões do algoritmo transgenético. Isso é observado pelos bons resultados das versões **cross0** e **plasmid0**. No primeiro experimento, dois algoritmos destacam-se: **cross0** e **gde**. Ambos utilizam o *crossover* diferencial proposto. No entanto, o **cross0** obtém excelente resultado em comparação ao **gde**. Isso é um forte indicativo da qualidade dos operadores da transgenética. Ademais, as versões **cross0** e **plasmid0** utilizam o conceito de subpopulações independentes. Neste aspecto, também há forte indicativo de que essa ideia ajudou a obter ótimos resultados. Isso se apoia no fato de que nos dois experimentos realizados as versões com subpopulações, **cross0** e **plasmid0**, obtêm excelentes resultados.

O terceiro modelo estudado foi proposto por Chen et al. (2013). Este modelo possui um elemento adicional que é a capacidade de um grupo ser atendido por mais de uma

fonte. O objetivo é maximizar a capacidade residual. Um algoritmo heurístico denominado MMVD foi proposto para resolução do problema. Este algoritmo tem como base a construção de diagramas de Voronoi para alocar as fontes de um grupo *multicast* da melhor maneira possível. Este algoritmo foi comparado a três algoritmos da literatura: O MMForrests proposto por Chen et al. (2013) e o P-MCF e E-MCF propostos por Guo et al. (2017). Os algoritmos P-MCF e E-MCF foram projetados originalmente para otimização do custo e consideram apenas uma sessão. Entretanto, é possível adaptá-los para considerar a otimização da capacidade residual e construir árvores para mais de uma sessão. Os algoritmos foram avaliados em um extenso experimento computacional cujos resultados foram testados estatisticamente. Três métricas foram consideradas: capacidade residual, custo e tempo de processamento. Em geral, o MMVD obteve resultados pelo menos tão bons quanto as demais heurísticas com relação ao valor de capacidade residual. Além disso, obteve excelentes resultados quanto ao custo de configuração e dominou praticamente todos os cenários considerando o tempo de processamento.

De forma resumida, o presente trabalho traz importantes contribuições para o estudo do problema de roteamento *multicast*. Dois novos modelos foram projetados. Vinte e três novos algoritmos foram projetados incluídos operadores e as meta-heurísticas. Banco de instâncias foram projetados, totalizando 2500 instâncias. Todas as avaliações conduzidas foram devidamente planejadas e recursos importantes foram utilizados como afinador de parâmetros e testes estatísticos para comparar as amostras obtidas por cada um dos algoritmos nos diferentes modelos. Como resultado, destacam-se os algoritmos ILS para o modelo *Modelo_Z*, a heurística MMVD para o problema com mono-objetivo com múltiplas fontes e o algoritmo transgenético com subpopulações.

8.1 Trabalhos Futuros

Algumas possibilidades de pesquisa podem ser conduzidas a partir dos resultados deste trabalho:

- aplicar as presentes ideias do algoritmo transgenético a outros problemas de otimização;
- validar o uso de subpopulações no transgenético aplicado a outros problemas de otimização;
- considerar o uso de soluções de elite no algoritmo transgenético sob diferentes condições;
- aplicar novos operadores de árvore de Steiner na criação de novos operadores transgenéticos;

- adaptar algoritmos de busca local para árvores de Steiner como *transposon* para ser usado no algoritmo transgenético proposto.
- estudar um modelo multiobjetivo com múltiplas fontes por sessão;
- considerar novos objetivos de otimização na modelagem multiobjetivo.

A Resultados do *modelo_Z*

Neste apêndice, todos os detalhes das comparações realizadas com os algoritmos propostos para o *Modelo_Z* são apresentados. Os algoritmos foram comparados com relação ao valor de capacidade residual, função-objetivo do problema. Além disso, também foi levado em conta o valor de custo das soluções obtidas. O modelo impõe um limite de custo, porém este valor pode ser reduzido pelo algoritmo até o limite inferior, conforme ilustrado nas Tabelas 9 e 10.

Os resultados são apresentados por meio dos gráficos presentes nas Figuras 26-33. Estes gráficos ilustram o valor mediano de capacidade residual e o tempo de processamento obtidos em cada instância considerando 50 execuções na avaliação dos algoritmos no parâmetro W . Além disso, as avaliações estatísticas dos resultados ilustrados nos gráficos estão presentes nas Tabelas 29-32. Estes resultados representam a avaliação do *Posthoc* de Nemenyi após verificação de diferença por meio de teste de Friedman. Ambos os testes estatísticos foram realizados com nível de significância de 0.05.

Os gráficos que apresentam o valor de custo não são ilustrados no texto. Isso se deve ao fato de não apresentarem visualmente as diferenças entre os algoritmos. Entretanto, os testes estatísticos apontam tais diferenças.

Por fim, na Tabela 33 são ilustrados os resultados da comparação entre os algoritmos utilizando o teste de Friedman e *Posthoc* de Nemenyi diante da variação do número de vértices da rede, $n \in \{30, 60, 120, 240\}$.

Figura 26 – Comparação dos algoritmos por Capacidade Residual no conjunto $n = 30$ por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.

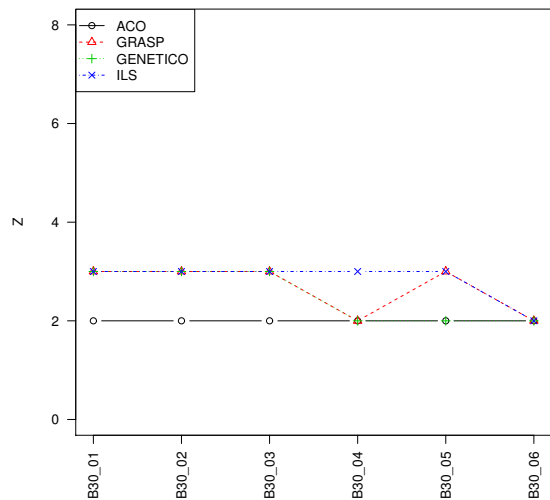
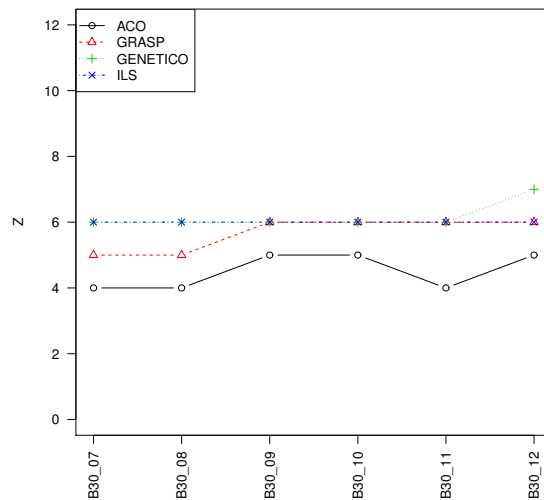
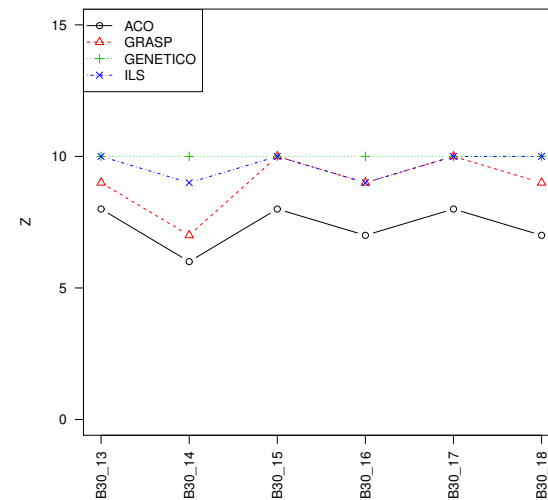
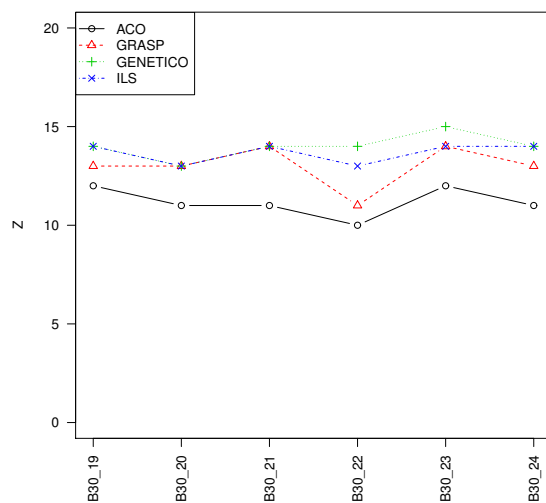
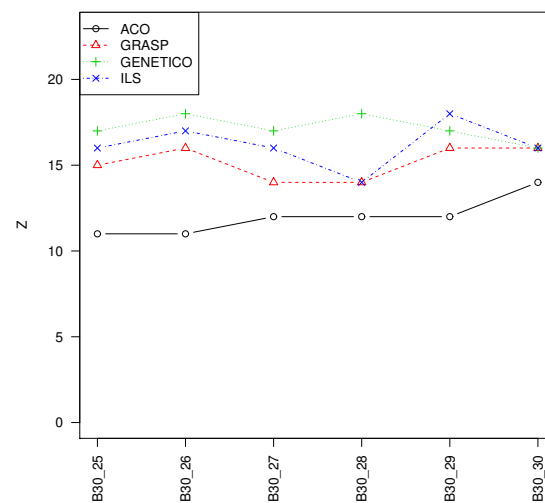
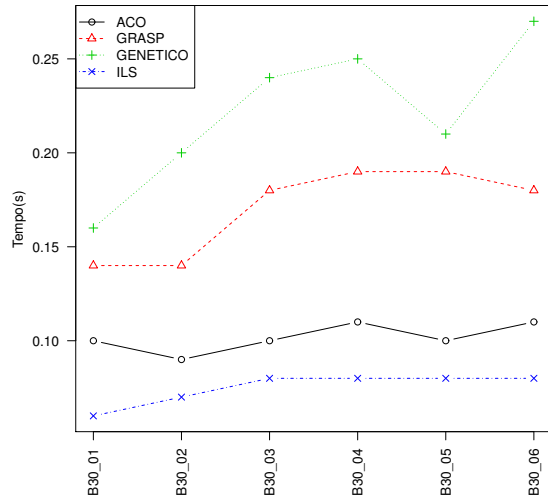
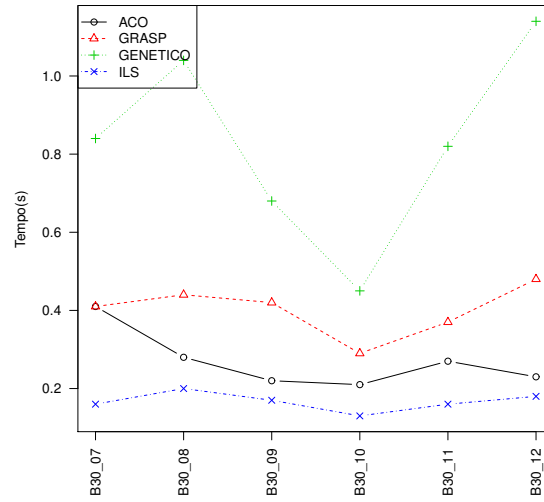
(a) $|W| = 5$ (b) $|W| = 10$ (c) $|W| = 15$ (d) $|W| = 20$ (e) $|W| = 25$ 

Figura 27 – Comparação dos algoritmos por Tempo de Processamento no conjunto com $n = 30$ por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.

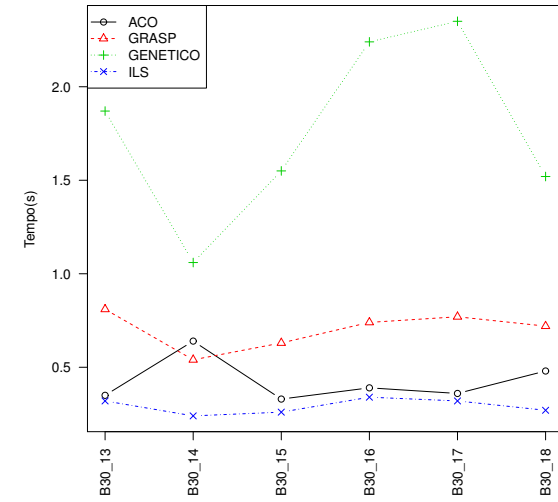
(a) $|W| = 5$



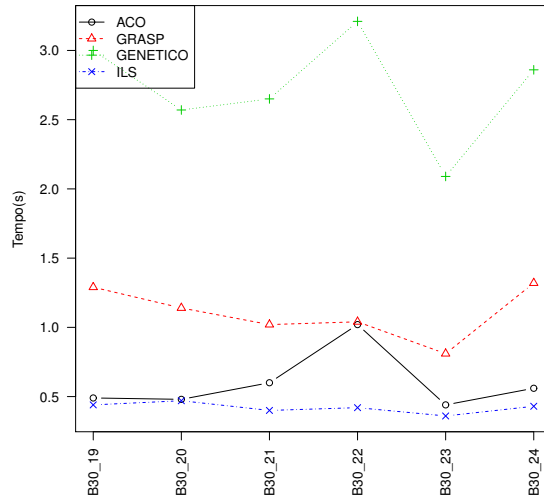
(b) $|W| = 10$



(c) $|W| = 15$



(d) $|W| = 20$



(e) $|W| = 25$

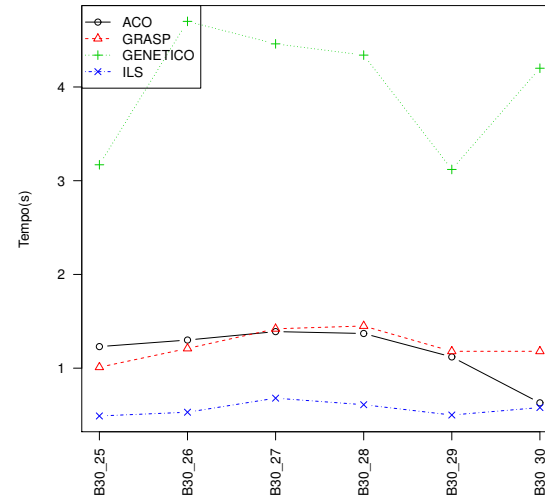
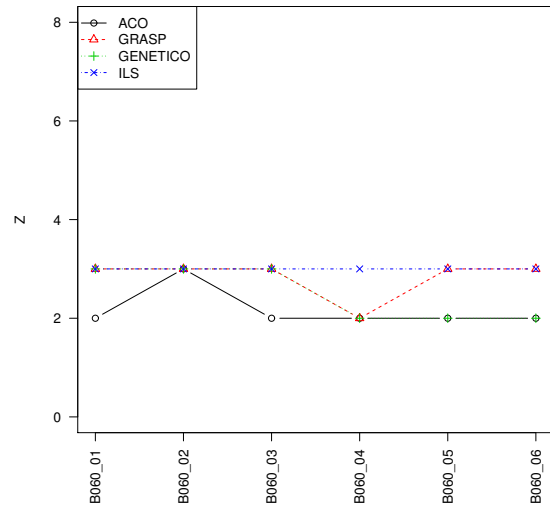


Tabela 29 – *Posthoc* de Nemenyi sobre resultado dos algoritmos por valor de W nas instâncias do conjunto com $n = 30$ utilizando nível de significância igual 0,05.

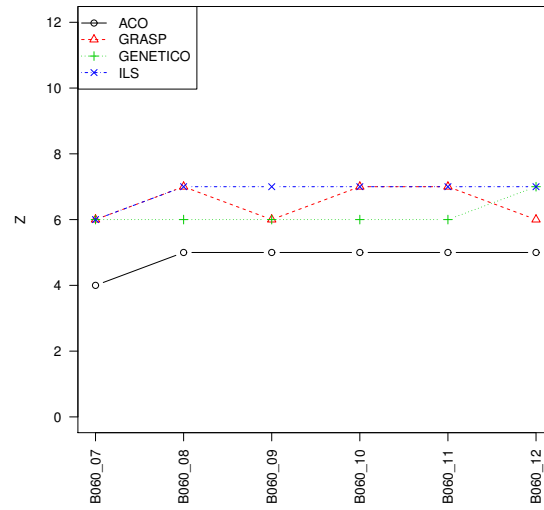
Capacidade Residual				Custo				Tempo			
$ W = 5$				$ W = 5$				$ W = 5$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,2800	-	-	GRASP	0,9700	-	-	GRASP	0,5363	-	-
GENETICO	0,5400	0,9700	-	GENETICO	0,5400	0,2800	-	GENETICO	0,0367	0,5363	-
ILS	0,1100	0,9700	0,8100	ILS	0,2800	0,1100	0,9700	ILS	0,5363	0,0367	0,0003
$ W = 10$				$ W = 10$				$ W = 10$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,1454	-	-	GRASP	0,9960	-	-	GRASP	0,6784	-	-
GENETICO	0,0095	0,7458	-	GENETICO	0,9960	0,9700	-	GENETICO	0,0497	0,4659	-
ILS	0,0267	0,9081	0,9870	ILS	0,0370	0,0660	0,0190	ILS	0,4659	0,0497	0,0003
$ W = 15$				$ W = 15$				$ W = 15$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,2277	-	-	GRASP	1	-	-	GRASP	0,8078	-	-
GENETICO	0,0044	0,4659	-	GENETICO	0,9700	1	-	GENETICO	0,0664	0,3986	-
ILS	0,0267	0,8078	0,9441	ILS	0,4000	0,2800	0,1800	ILS	0,3986	0,0664	0,0003
$ W = 20$				$ W = 20$				$ W = 20$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,2277	-	-	GRASP	0,5400	-	-	GRASP	0,5363	-	-
GENETICO	0,0044	0,4659	-	GENETICO	1	0,6800	-	GENETICO	0,0367	0,5363	-
ILS	0,0267	0,8078	0,9441	ILS	0,1800	0,9100	0,2800	ILS	0,5363	0,0367	0,0003
$ W = 25$				$ W = 25$				$ W = 25$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,3360	-	-	GRASP	0,6784	-	-	GRASP	0,9702	-	-
GENETICO	0,0020	0,2280	-	GENETICO	0,8078	0,9961	-	GENETICO	0,1136	0,2786	-
ILS	0,0270	0,6780	0,8620	ILS	0,0044	0,1136	0,0664	ILS	0,2786	0,1136	0,0003

Figura 28 – Comparação dos algoritmos por Capacidade Residual no conjunto $n = 60$ por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.

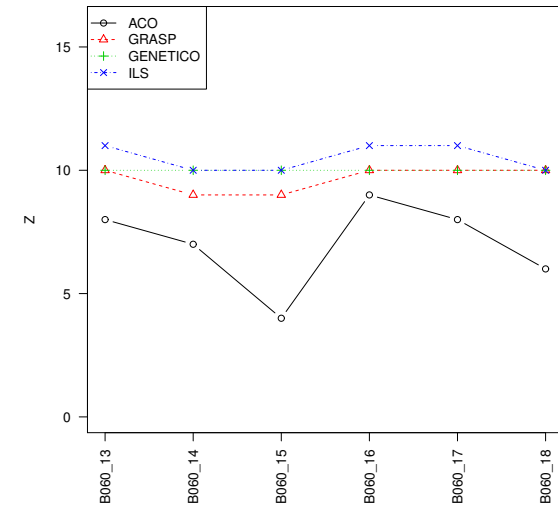
(a) $|W| = 5$



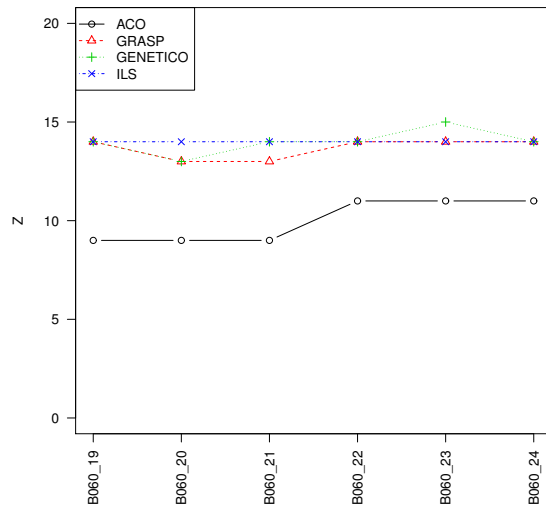
(b) $|W| = 10$



(c) $|W| = 15$



(d) $|W| = 20$



(e) $|W| = 25$

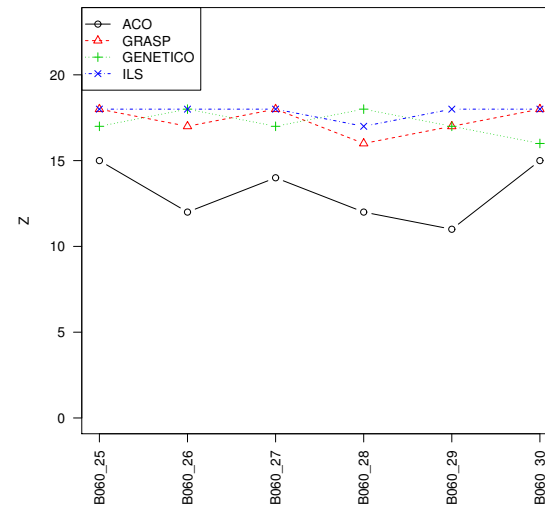


Figura 29 – Comparação dos algoritmos por Tempo de Processamento no conjunto com $n = 60$ e por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.

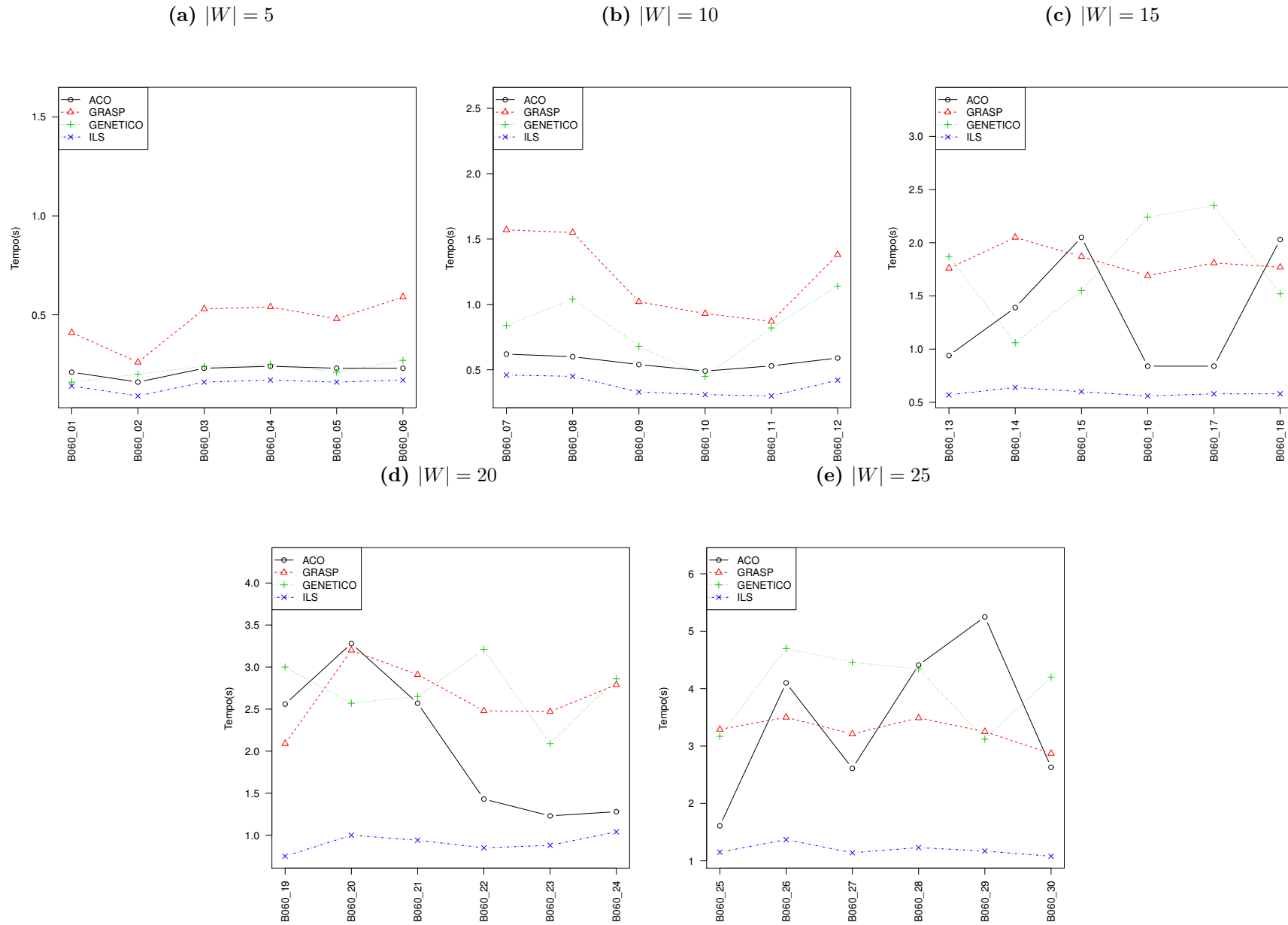


Tabela 30 – *PostHoc* de Nemenyi sobre resultado dos algoritmos por valor de W nas instâncias do conjunto com $n = 60$ e nível de significância igual a 0,05.

Capacidade Residual				Custo				Tempo			
$ W = 5$				$ W = 5$				$ W = 5$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,2800	-	-	GRASP	1	-	-	GRASP	0,1136	-	-
GENETICO	0,8100	0,8100	-	GENETICO	0,6780	0,6780	-	GENETICO	0,9702	0,2786	-
ILS	0,1100	0,9700	0,5400	ILS	0,0660	0,0660	0,5360	ILS	0,2786	0,0003	0,1136
$ W = 10$				$ W = 10$				$ W = 10$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,0366	-	-	GRASP	1	-	-	GRASP	0,0664	-	-
GENETICO	0,1833	0,9081	-	GENETICO	0,5400	0,4000	-	GENETICO	0,8078	0,3986	-
ILS	0,0044	0,9081	0,5363	ILS	0,9100	0,8100	0,9100	ILS	0,3986	0,0003	0,0664
$ W = 15$				$ W = 15$				$ W = 15$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,2280	-	-	GRASP	0,1100	-	-	GRASP	0,9700	-	-
GENETICO	0,0500	0,9080	-	GENETICO	0,6800	0,6800	-	GENETICO	0,9960	0,9960	-
ILS	0,0020	0,3360	0,7460	ILS	0,1800	1	0,8100	ILS	0,0660	0,0190	0,0370
$ W = 20$				$ W = 20$				$ W = 20$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,1140	-	-	GRASP	0,5360	-	-	GRASP	0,8078	-	-
GENETICO	0,0190	0,9080	-	GENETICO	0,0660	0,6780	-	GENETICO	0,6784	0,9961	-
ILS	0,0190	0,9080	1	ILS	0,0660	0,6780	1	ILS	0,1833	0,0191	0,0095
$ W = 25$				$ W = 25$				$ W = 25$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,0664	-	-	GRASP	0,3986	-	-	GRASP	1	-	-
GENETICO	0,1136	0,9961	-	GENETICO	0,1136	0,9081	-	GENETICO	0,9081	0,9081	-
ILS	0,0044	0,8078	0,6784	ILS	0,0044	0,2786	0,6784	ILS	0,0664	0,0664	0,0095

Figura 30 – Comparação dos algoritmos por Capacidade Residual no conjunto com $n = 120$ por valor de W . Os resultados corresponde a média de 30 execuções em cada instância.

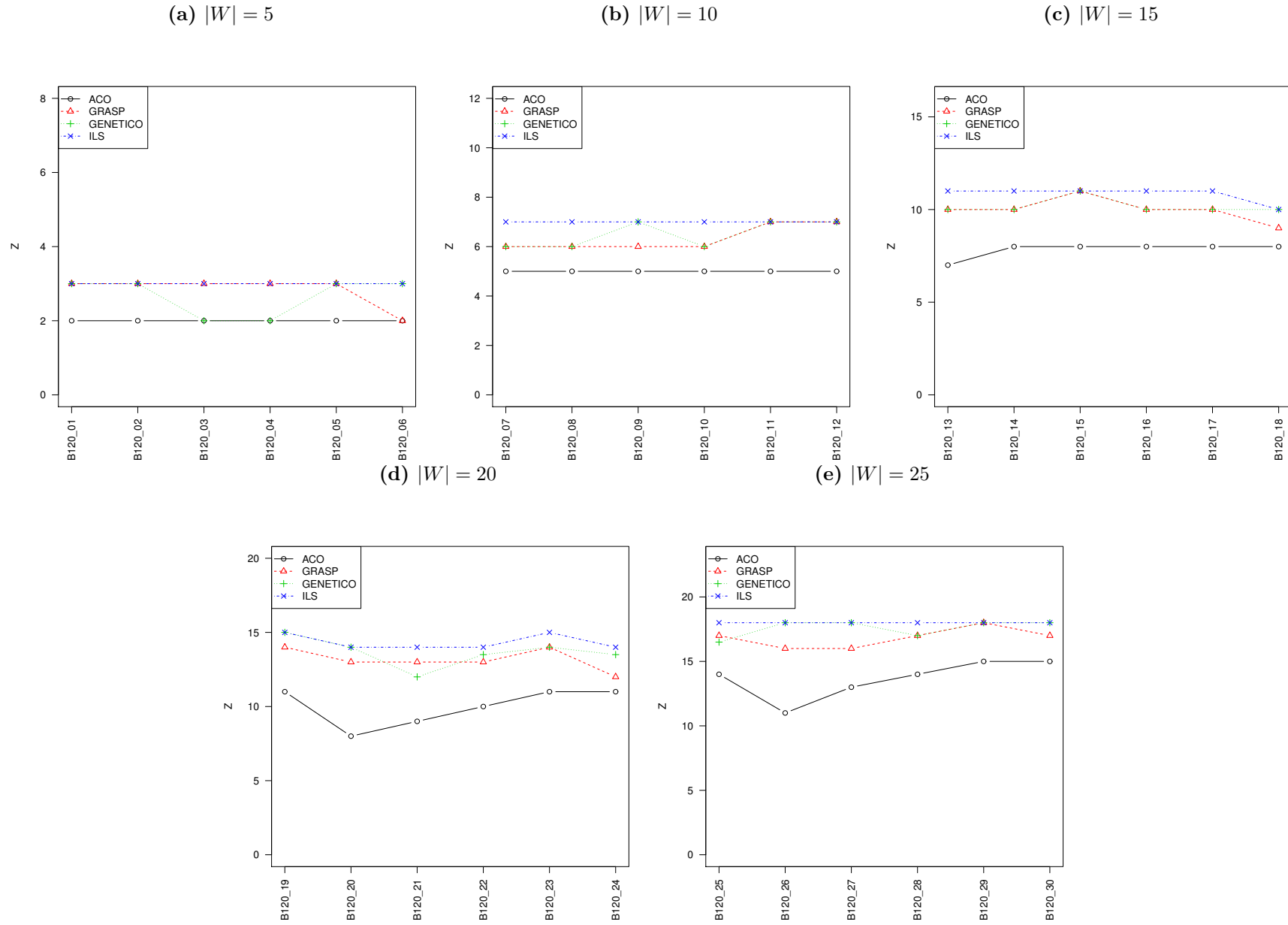
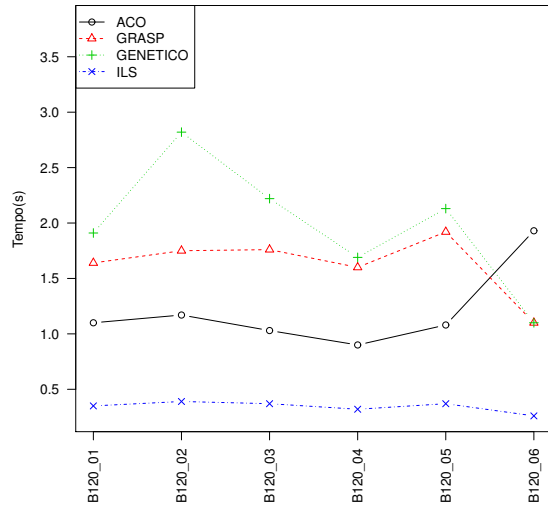
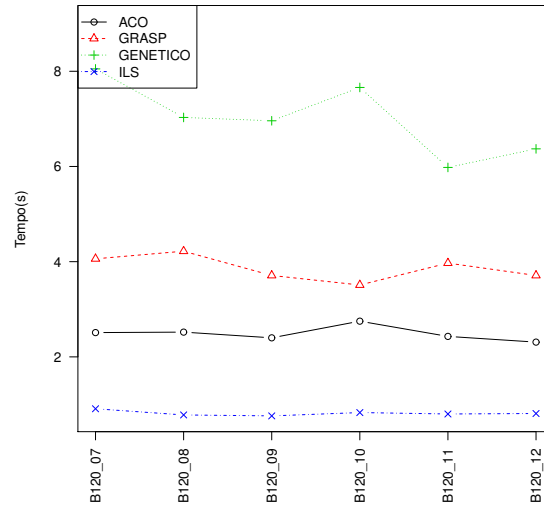


Figura 31 – Comparação dos algoritmos por Tempo de Processamento no conjunto com $n = 120$ e por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.

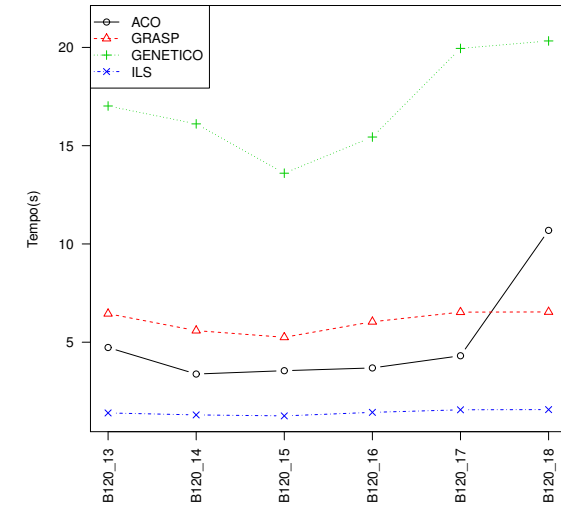
(a) $|W| = 5$



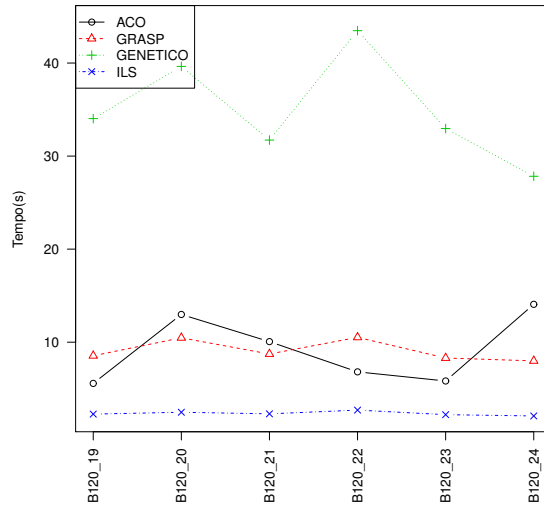
(b) $|W| = 10$



(c) $|W| = 15$



(d) $|W| = 20$



(e) $|W| = 25$

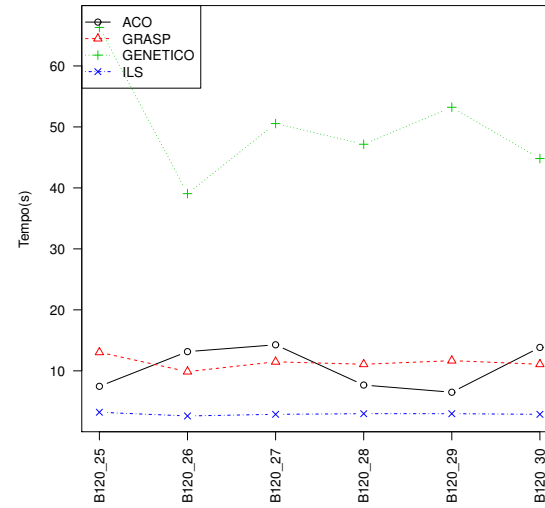
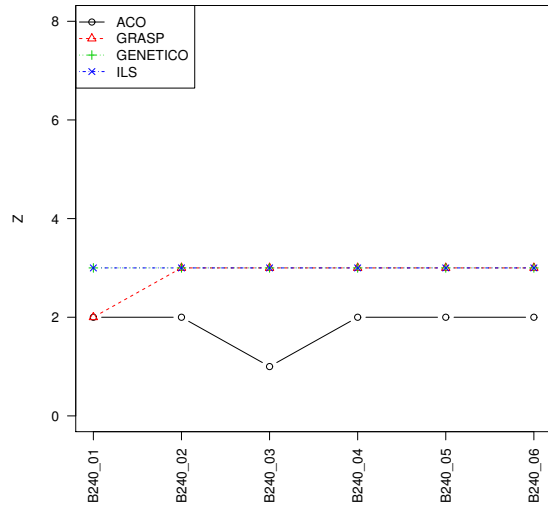


Tabela 31 – *PostHoc* de Nemenyi sobre resultado dos algoritmos por valor de W nas instâncias do conjunto com $n = 120$ e nível de significância igual a 0,05.

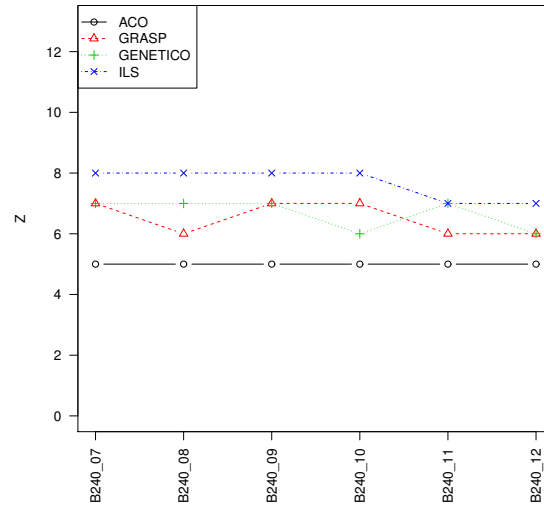
Capacidade Residual				Custo				Tempo			
$ W = 5$				$ W = 5$				$ W = 5$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,1140	-	-	GRASP	0,9080	-	-	GRASP	0,8624	-	-
GENETICO	0,2790	0,9700	-	GENETICO	0,5360	0,1830	-	GENETICO	0,2277	0,6784	-
ILS	0,0370	0,9700	0,8080	ILS	0,0190	0,0020	0,3990	ILS	0,2786	0,0497	0,0013
$ W = 10$				$ W = 10$				$ W = 10$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,1450	-	-	GRASP	0,9080	-	-	GRASP	0,5363	-	-
GENETICO	0,0660	0,9870	-	GENETICO	0,0660	0,2790	-	GENETICO	0,0367	0,5363	-
ILS	0,0030	0,5360	0,7460	ILS	0,1140	0,3990	0,9960	ILS	0,5363	0,0367	0,0003
$ W = 15$				$ W = 15$				$ W = 15$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,1833	-	-	GRASP	0,9960	-	-	GRASP	0,8078	-	-
GENETICO	0,0875	0,9870	-	GENETICO	0,0190	0,0370	-	GENETICO	0,0664	0,3986	-
ILS	0,0013	0,3358	0,5363	ILS	0,1140	0,1830	0,9080	ILS	0,3986	0,0664	0,0003
$ W = 20$				$ W = 20$				$ W = 20$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,3358	-	-	GRASP	1	-	-	GRASP	1	-	-
GENETICO	0,0497	0,8078	-	GENETICO	0,0660	0,0660	-	GENETICO	0,1833	0,1833	-
ILS	0,0008	0,1454	0,6078	ILS	0,1830	0,1830	0,9700	ILS	0,1833	0,1833	0,0003
$ W = 25$				$ W = 25$				$ W = 25$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,2280	-	-	GRASP	0,8080	-	-	GRASP	1	-	-
GENETICO	0,0370	0,8620	-	GENETICO	0,1830	0,0190	-	GENETICO	0,1833	0,1833	-
ILS	0,0030	0,3990	0,8620	ILS	0,3990	0,0660	0,9700	ILS	0,1833	0,1833	0,0003

Figura 32 – Comparação dos algoritmos por Capacidade Residual com $n = 240$ e por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.

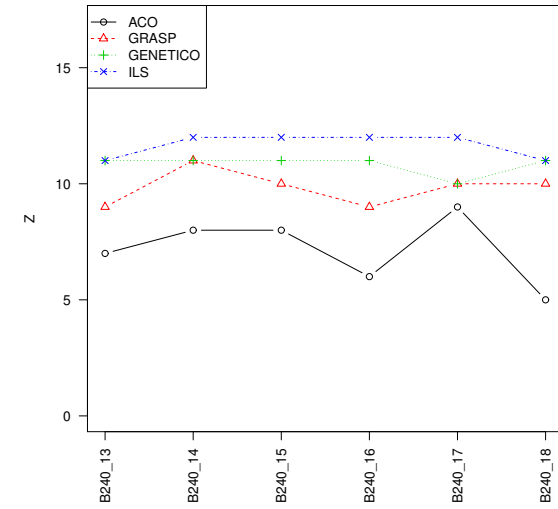
(a) $|W| = 5$



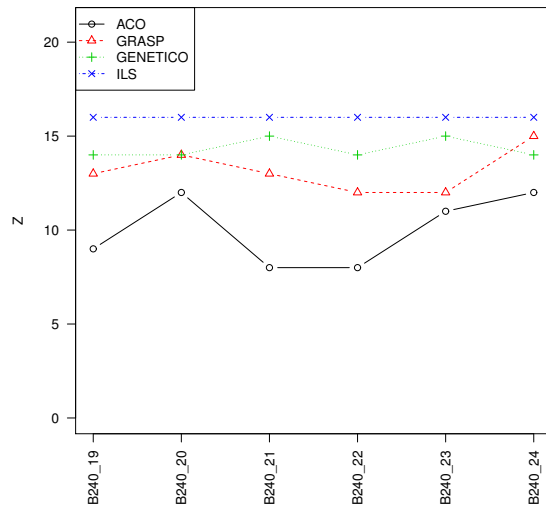
(b) $|W| = 10$



(c) $|W| = 15$



(d) $|W| = 20$



(e) $|W| = 25$

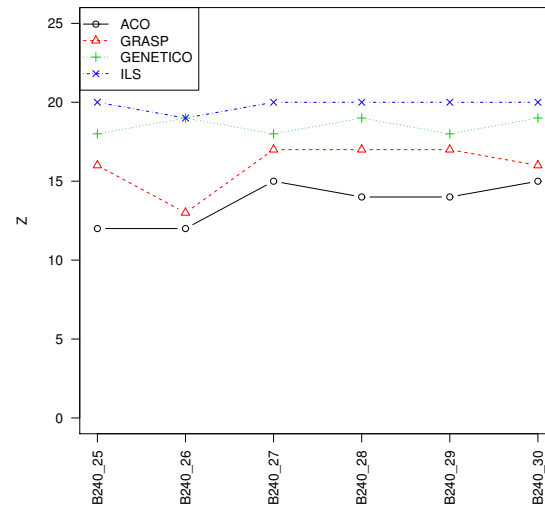


Figura 33 – Comparação dos algoritmos por Tempo de Processamento com $n = 240$ e por valor de W . Os resultados correspondem a média de 30 execuções em cada instância.

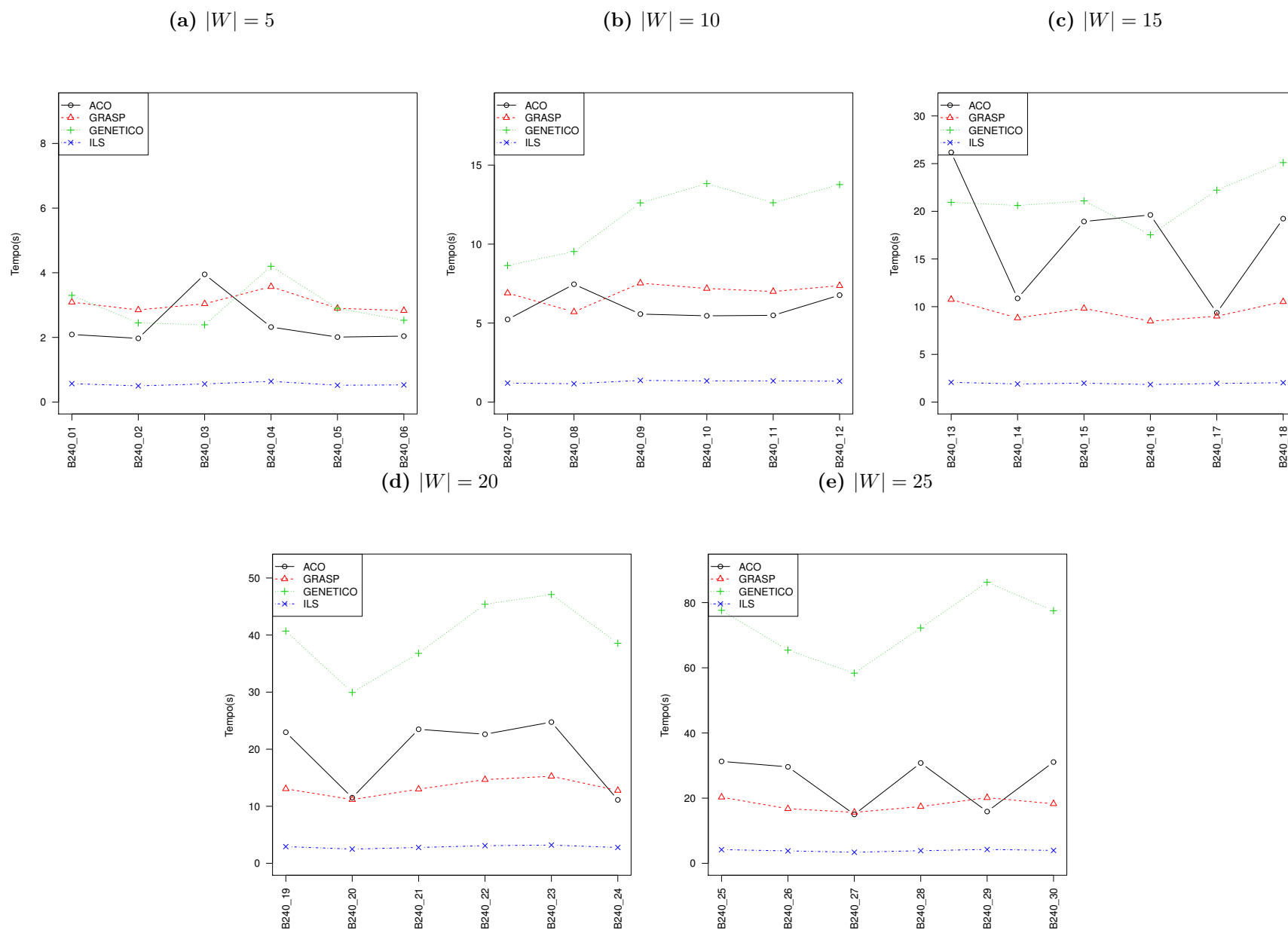


Tabela 32 – *PostHoc* de Nemenyi sobre resultado dos algoritmos por valor de W nas instâncias do conjunto 1 com $n = 240$ e nível de significância igual a 0,05.

Capacidade Residual				Custo				Tempo			
$ W = 5$				$ W = 5$				$ W = 5$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,1140	-	-	GRASP	0,9702	-	-	GRASP	0,4659	-	-
GENETICO	0,0370	0,9700	-	GENETICO	0,2786	0,1136	-	GENETICO	0,6078	0,9961	-
ILS	0,0370	0,9700	1	ILS	0,0095	0,0020	0,5363	ILS	0,2786	0,0065	0,0135
$ W = 10$				$ W = 10$				$ W = 10$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,2277	-	-	GRASP	1	-	-	GRASP	0,8078	-	-
GENETICO	0,1136	0,9870	-	GENETICO	0,0370	0,0370	-	GENETICO	0,0664	0,3986	-
ILS	0,0005	0,1833	0,3358	ILS	0,2790	0,2790	0,8080	ILS	0,3986	0,0664	0,0003
$ W = 15$				$ W = 15$				$ W = 15$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,3986	-	-	GRASP	0,5363	-	-	GRASP	0,2786	-	-
GENETICO	0,0367	0,6784	-	GENETICO	0,0008	0,0664	-	GENETICO	0,9702	0,1136	-
ILS	0,0008	0,1136	0,6784	ILS	0,0191	0,3986	0,8078	ILS	0,0095	0,5363	0,0020
$ W = 20$				$ W = 20$				$ W = 20$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,3358	-	-	GRASP	0,5363	-	-	GRASP	0,8078	-	-
GENETICO	0,0875	0,9081	-	GENETICO	0,0003	0,0367	-	GENETICO	0,3986	0,0664	-
ILS	0,0003	0,0875	0,3358	ILS	0,0367	0,5363	0,5363	ILS	0,0664	0,3986	0,0003
$ W = 25$				$ W = 25$				$ W = 25$			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,5363	-	-	GRASP	0,9961	-	-	GRASP	0,9702	-	-
GENETICO	0,0267	0,4659	-	GENETICO	0,0095	0,0191	-	GENETICO	0,2786	0,1136	-
ILS	0,0005	0,0497	0,6784	ILS	0,1833	0,2786	0,6784	ILS	0,1136	0,2786	0,0003

Tabela 33 – *PostHoc* de Nemenyi aplicado aos resultados obtidos na comparação dos algoritmos diante da variação no tamanho da rede com $n \in \{30, 60, 120, 240\}$ e nível de significância igual a 0,05.

Capacidade Residual				Custo				Tempo			
n = 30				n = 30				n = 30			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,0002	-	-	GRASP	0,80483	-	-	GRASP	0,0980	-	-
GENETICO	2,50E-10	0,0771	-	GENETICO	0,85444	0,9996	-	GENETICO	9,60E-08	0,0045	-
ILS	1,20E-08	0,2733	0,9321	ILS	2,00E-06	0,0002	0,0001	ILS	0,0045	9,60E-08	2,70E-14
n = 60				n = 60				n = 60			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	4,40E-06	-	-	GRASP	0,1232	-	-	GRASP	0,0350	-	-
GENETICO	1,20E-05	1	-	GENETICO	0,0012	0,4375	-	GENETICO	0,4370	0,6270	-
ILS	3,10E-11	0,2100	0,1400	ILS	5,70E-06	0,0350	0,6268	ILS	2,50E-05	1,80E-12	6,40E-09
n = 120				n = 120				n = 120			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	5,10E-05	-	-	GRASP	0,99	-	-	GRASP	0,53096	-	-
GENETICO	6,90E-07	0,8304	-	GENETICO	3,40E-06	6,90E-07	-	GENETICO	4,40E-06	0,00181	-
ILS	1,50E-13	0,0075	0,0870	ILS	5,70E-06	1,20E-06	1	ILS	0,00056	9,10E-07	2,90E-14
n = 240				n = 240				n = 240			
	ACO	GRASP	GENETICO		ACO	GRASP	GENETICO		ACO	GRASP	GENETICO
GRASP	0,69	-	-	GRASP	0,69	-	-	GRASP	0,9853	-	-
GENETICO	3,10E-11	4,00E-08	-	GENETICO	3,10E-11	4,00E-08	-	GENETICO	0,0045	0,0012	-
ILS	1,30E-07	4,00E-05	0,5600	ILS	1,30E-07	4,00E-05	0,5600	ILS	3,40E-06	2,00E-05	4,20E-14

B Resultados do *Modelo_{chen}*

Neste apêndice, os resultados obtidos pela heurística MMVD em comparação com as heurísticas MMForests, P-MCF e E-MCF são apresentados com todos os detalhes. A avaliação foi realizada nos blocos de instâncias definidos na Tabela 7. Cada bloco de instância está associado a um cenário de experimentação e considera os seguintes parâmetros $n, d = |D^k|, r = |W|$ e $s = |S_k|$.

Os cenários são resumidos a seguir:

- **Cenário 1** - Multissessão com requerimento de tráfego $t^k = 1$ e número de fontes $|S_k| = 1$;
- **Cenário 2** - Multissessão com requerimento de tráfego $t^k = 1$ e número de fontes $|S_k| > 1$;
- **Cenário 3** - Multissessão com requerimento de tráfego $t^k \geq 1$ e número de fontes $|S_k| \geq 1$;
- **Cenário 4** - Avaliação de escalabilidade.

As medianas das execuções dos algoritmos em cada um dos cenários 1, 2, 3 e 4 são ilustradas nas Tabelas 34, 36, 38 e 40, respectivamente. Cada tabela é dividida pelo parâmetro de configuração avaliado, r, d, n e s . Para cada algoritmo avaliado, três colunas foram adicionadas: Z , custo e tempo de processamento. O valor ilustrado como resultado é a mediana da execução de cada heurística em um conjunto de 30 instâncias para cada configuração. Por exemplo, na Tabela 34 onde $r = 20$ há 30 instâncias e os 30 resultados produzem a mediana para cada heurística, 17 para o MMVD, 16 para o P-MCF, 16 para E-MCF e 16 para o MMForests. Além disso, esses resultados formam a amostra analisada nos testes estatísticos. Os valores 0 no tempo de processamento indicam que o valor obtido foi menor que 0,01 segundo.

Os resultados dos testes de Friedman e *posthoc* de Shaffer nos cenários 1, 2, 3 e 4 são apresentados nas Tabelas 35, 37, 39 e 41. Os valores ilustrados nas tabelas são referentes ao p-value obtido após o teste de *posthoc* de Shaffer. Neste caso, é avaliado se houve diferença entre o MMVD e os demais procedimentos no cenário/parâmetro, r, d e n , considerados. Além disso, o resultado é apresentado para capacidade residual, custo e tempo de processamento.

Tabela 34 – Mediana dos resultados obtidos no cenário 1. Cada valor de parâmetro possui 30 instâncias e a mediana corresponde ao resultado do algoritmo nesses instâncias.

Cenário 1 - $t^k = 1, s = 1$												
r	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
20	17	1250,5	0	16	1403	0,04	16	1386	0,04	16	1243,5	0
40	34	2540	0	34	2824	0,09	34	2808	0,09	34	2516,5	0,01
60	52	3819	0,01	52	4252,5	0,13	52	4228,5	0,14	52	3798	0,01
80	70	5106	0,01	69	5677,5	0,18	69	5633,5	0,19	69	5083,5	0,02
100	80	6400,5	0,02	87	7116,5	0,23	87	7039	0,24	87	6348	0,03

d	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
10	37	1172	0	37	1279	0,02	37	1308,5	0,02	37	1160	0
20	35	1925	0	35	2182,5	0,05	35	2151	0,05	35	1921,5	0,01
30	34	2540	0	34	2824	0,09	34	2808	0,09	34	2516,5	0,01
40	33	3043	0	33	3374,5	0,14	33	3334,5	0,15	33	3038,5	0,01
50	33	3501	0	32	3865,5	0,22	32	3796,5	0,24	32	3508,5	0,01

n	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
40	27	1308	0	28	1330	0,04	28	1327	0,04	27	1306,5	0
80	31	1739	0	31	1844,5	0,05	31	1837,5	0,06	31	1737,5	0
120	33	2058,5	0	33	2220,5	0,06	33	2216,5	0,07	33	2055,5	0
160	34	2302,5	0	33	2533	0,08	34	2512,5	0,08	34	2295	0,01
200	34	2540	0	34	2824	0,09	34	2808	0,09	34	2516,5	0

Tabela 35 – Resultado do teste Post-hoc de Shaffer sobre as amostras obtidas pelos algoritmos no cenário 1 e parâmetros r, d e n . O nível de significância utilizado foi 0,05.

Cenário 1 - $t^k = 1, s = 1$									
r	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
20	0,6575915	1	1	5,91953E-09	1,98945E-08	1	5,92E-04	1,35E-13	0,008099388
40	1	1	1	8,04427E-09	7,81459E-07	0,736240501	5,92E-04	1,35E-13	0,008099388
60	1	1	1	8,04427E-09	1,32543E-06	0,634621016	5,92E-04	1,35E-13	0,008099388
80	2,53495E-05	0,000190027	1	2,4096E-10	0,000123945	0,161513318	5,92E-04	1,35E-13	0,008099388
100	1,60938E-05	0,000123945	1	5,91953E-09	3,24753E-05	0,219197167	1,69E-04	8,21E-13	0,005399592
d	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
10	0,951931524	0,6575915	1	4,75997E-06	1,98945E-08	0,323026637	1,23E-10	2,68E-16	0,005399592
20	0,6575915	0,690418021	1	1,53578E-11	5,97866E-07	0,841480581	5,92E-09	1,35E-18	0,008099388
30	1	1	1	8,04427E-09	7,81459E-07	0,736240501	1,69E-09	8,21E-18	0,005399592
40	0,951931524	1	0,484539955	3,13857E-11	1,71991E-06	0,841480581	1,69E-09	8,21E-18	0,005399592
50	0,001909303	0,002021576	0,136500792	3,61275E-12	4,75997E-06	1	4,66E-10	4,78E-17	0,005399592
n	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
40	0,020801843	0,009796228	1	8,69075E-09	1,01896E-06	0,960122388	1,69E-09	8,21E-18	0,005399592
80	1	1	1	6,71101E-10	4,81344E-08	0,920344325	5,92E-09	1,35E-18	0,008099388
120	1	1	1	8,04427E-09	4,81344E-08	1	3,18E-09	3,35E-18	0,005805619
160	0,431163829	0,484539955	1	3,12015E-11	1,01896E-06	1	5,92E-09	1,35E-18	0,008099388
200	1	1	1	8,04427E-09	7,81459E-07	0,736240501	1,69E-09	8,21E-18	0,005399592

Tabela 36 – Mediana dos resultados obtidos no cenário 2. Cada valor de parâmetro possui 30 instâncias e a mediana corresponde ao resultado do algoritmo nesses instâncias.

Cenário 2 - $t^k = 1, s \geq 1$												
r	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
20	15	1327	0	15	1471,5	0,05	15	1465,5	0,05	15	1402,5	0,01
40	32	2689,5	0	31	2942	0,11	31	2927,5	0,11	31	2860	0,02
60	48	4040	0,01	47	4428,5	0,16	47	4403,5	0,17	47	4280	0,04
80	65	5414,5	0,01	64	5905	0,22	64	5898	0,22	64	5741	0,05
100	81	6770	0,02	80	7379,5	0,27	80	7358	0,28	80	7189,5	0,06

d	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
10	35	1267	0	35	1449	0,03	35	1467,5	0,03	35	1404	0,01
20	33	2070,5	0	33	2314,5	0,06	33	2300	0,06	33	2213	0,02
30	32	2689,5	0	31	2942	0,11	31	2927,5	0,11	31	2860	0,02
40	30	3200,5	0	30	3476	0,17	30	3458	0,18	30	3406	0,03
50	29	3678	0	29	3918,5	0,27	29	3890,5	0,27	29	3878	0,03

n	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
40	22	1276	0	22	1285,5	0,05	22	1286,5	0,05	21	1365	0,01
80	27	1766,5	0	27	1846,5	0,07	27	1836,5	0,07	27	1882,5	0,01
120	29	2143,5	0	29	2280,5	0	29	2279	0,08	29	2298,5	0,01
160	31	2415	0	30	2621	0,09	30	2619,5	0,1	30	2590	0,02
200	32	2689,5	0	31	2942	0,11	31	2927,5	0,11	31	2860	0,02

Tabela 37 – Resultado do teste Post-hoc de Shaffer sobre as amostras obtidas pelos algoritmos no cenário 2 e parâmetros r, d e n . O nível de significância utilizado foi 0,05.

Cenário 2 - $t^k = 1, s \geq 1$									
r	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
20	1	1	1	2,61155E-13	2,2155E-11	0,000954652	5,91953E-09	1,35431E-18	0,008099388
40	0,000190027	0,000160149	0,049185216	1,67342E-14	1,56008E-11	0,002748552	5,91953E-09	1,35431E-18	0,008099388
60	6,49505E-05	0,000190027	0,002021576	1,44232E-15	1,23347E-10	0,002748552	5,91953E-09	1,35431E-18	0,008099388
80	6,94816E-07	1,71991E-06	2,0386E-05	1,44232E-15	6,25259E-11	0,003870413	5,91953E-09	1,35431E-18	0,008099388
100	1,29189E-09	1,13937E-07	7,81459E-07	3,29755E-15	1,23347E-10	0,002900545	1,69389E-09	8,20897E-18	0,005399592
d	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
10	0,969079911	0,969079911	1	3,3555E-10	1,77678E-13	0,000234454	1,23347E-10	2,67887E-16	0,005399592
20	0,16684137	0,215581915	0,16684137	1,77678E-13	8,63303E-13	0,003870413	5,91953E-09	1,35431E-18	0,008099388
30	0,000190027	0,000160149	0,049185216	1,67342E-14	1,56008E-11	0,002748552	1,69389E-09	8,20897E-18	0,005399592
40	0,431163829	0,431163829	0,431163829	1,67342E-14	3,18205E-09	0,000190027	1,69389E-09	8,20897E-18	0,005399592
50	0,16684137	0,690418021	0,16684137	3,82907E-13	1,99923E-07	1,01896E-06	4,66131E-10	4,78297E-17	0,005399592
n	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
40	1	1	0,042856864	0,000288578	0,000288578	4,78297E-17	1,69389E-09	8,20897E-18	0,005399592
80	0,16684137	0,690418021	0,16684137	2,6773E-08	0,000354353	2,67887E-16	5,91953E-09	1,35431E-18	0,008099388
120	0,020801843	0,017878579	0,004848694	2,32449E-09	7,81459E-07	1,25052E-10	3,18205E-09	3,3508E-18	0,005805619
160	0,000868177	0,000954652	0,000954652	3,61275E-12	3,3555E-10	4,08413E-05	5,91953E-09	1,35431E-18	0,008099388
200	0,000190027	0,000160149	0,049185216	1,67342E-14	1,56008E-11	0,002748552	1,69389E-09	8,20897E-18	0,005399592

Tabela 38 – Mediana dos resultados obtidos no cenário 3. Cada valor de parâmetro possui 30 instâncias e a mediana corresponde ao resultado do algoritmo nesses instâncias.

Cenário 3 - $t^k \geq 1$, $s \geq 1$												
r	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
20	78	1317	0	79,5	1445,5	0,06	80	1430	0,06	77	1446,5	0,01
40	67,5	2685	0	67	2937,5	0,12	68	2917,5	0,13	62,5	3006	0,03
60	58	4038	0,01	57	4409	0,19	57	4411,5	0,19	50	4546	0,04
80	45,5	5377,5	0,01	44,5	5918,5	0,25	44	5906,5	0,26	38,5	6028	0,06
100	35	6700	0,02	33,5	7383,5	0,32	32,5	7417	0,33	26	7587	0,08

d	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
10	42	948	0	42	1067,5	0,03	42	1067	0,03	40,5	1023	0,22
20	38	1551	0	37	1714,5	0,05	37	1697,5	0,06	34	1686	0,01
30	33	2023,5	0	33	2181	0,09	33	2166,5	0,09	29	2225,5	0,02
40	32	2409,5	0	30,5	2594,5	0,14	30,5	2601	0,15	28	2644	0,02
50	27	2772,5	0	25	2931	0,22	24	2907,5	0,22	21,5	3039,5	0,03

n	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
40	21,5	1257	0	20,5	1288	0,06	21,5	1286	0,06	14,5	1441	0,01
80	38	1789	0	36	1845,5	0,08	37	1846,5	0,08	33	1990,5	0,01
120	44	2160	0	43,5	2272,5	0,09	44	2264	0,09	40	2428,5	0,02
160	45,5	2464,5	0	45	2608	0,11	44	2615	0,11	40	2712	0,02
200	49	2726	0	49	2935,5	0,12	49	2930	0,13	44	3016	0,03

s	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
10	41	33897	0,2	39,5	36379	257,14	40	36064	232,66	34	38291	4,4
20	41,5	33610,5	0,18	40	36005,5	282,96	41,5	35830	252,93	33,5	38021	8,69
30	42	33229	0,18	39	35790	311,9	40	35586,5	274,38	33	37550	12,06
40	43,5	32971	0,17	43	35441	340,79	42,5	35296	296,68	35,5	37545	15
50	42,5	32708	0,16	40,5	35248	370,08	41	35013,5	321,89	34,5	37565,5	17,85

Tabela 39 – Resultado do teste Post-hoc de Shaffer sobre as amostras obtidas pelos algoritmos no cenário 3 e parâmetros r, d, n e s . O nível de significância utilizado foi 0,05.

Cenário 3 - $t^k \geq 1$, $s \geq 1$									
r	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
20	0,595720335	0,595720335	0,00020638	5,35104E-06	9,37615E-05	5,15862E-08	2,12697E-07	4,17316E-11	0,028611757
40	1	1	7,87934E-06	9,76526E-06	0,000439889	2,52337E-09	2,89007E-06	1,20294E-12	0,042917635
60	0,150130564	0,540688281	5,15862E-08	1,75652E-05	0,000439889	1,14402E-09	4,17368E-07	1,75795E-11	0,028611757
80	0,654373756	0,227258063	3,58324E-08	9,37615E-05	0,000439889	9,76188E-11	4,17368E-07	1,75795E-11	0,028611757
100	0,006598941	0,014101458	9,76188E-11	0,000266548	0,000159224	9,76188E-11	1,53845E-06	2,98468E-12	0,030337002
d	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
10	0,172821466	0,129892397	0,036144851	0,004399294	0,004352585	5,11109E-10	2,89007E-06	1,20294E-12	0,042917635
20	1	1	6,2286E-05	4,17368E-07	0,00071569	5,15862E-08	2,89007E-06	1,20294E-12	0,042917635
30	0,396898607	0,33404066	0,000343031	7,24179E-06	0,000343031	5,48478E-09	2,89007E-06	1,20294E-12	0,042917635
40	0,733870815	0,851966233	5,96982E-07	9,37615E-05	3,1143E-05	5,48478E-09	1,53845E-06	2,98468E-12	0,030337002
50	1	1	0,000108854	3,1143E-05	0,001815517	4,17316E-11	1,53845E-06	2,98468E-12	0,030337002
n	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
40	0,259232199	0,259232199	0,004352585	0,001446229	0,001446229	2,98468E-12	1,06826E-07	9,76188E-11	0,028611757
80	0,654373756	0,533728785	0,000159224	0,000266548	0,00071569	7,29711E-12	1,53845E-06	2,98468E-12	0,030337002
120	1	1	5,78014E-06	0,000439889	0,000439889	7,29711E-12	8,07156E-07	7,29711E-12	0,028611757
160	0,595347911	0,951170269	0,000122404	0,000266548	0,000266548	4,17316E-11	2,89007E-06	1,20294E-12	0,042917635
200	1	1	4,12446E-05	2,89007E-06	5,44268E-05	2,13653E-07	2,89007E-06	1,20294E-12	0,042917635
s	Capacidade Residual			Custo			Tempo de Processamento		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
10	0,851966233	0,851966233	2,11244E-06	1,53845E-06	0,030337002	2,98468E-12	1,20294E-12	2,89007E-06	0,042917635
20	0,476987727	0,806495941	3,02024E-07	9,76526E-06	0,021152187	1,20294E-12	1,20294E-12	2,89007E-06	0,042917635
30	1	1	7,15646E-05	5,35104E-06	0,002830935	2,25017E-10	1,20294E-12	2,89007E-06	0,042917635
40	1	1	1,31206E-05	9,76526E-06	0,021152187	1,20294E-12	1,20294E-12	2,89007E-06	0,042917635
50	1	1	7,24179E-06	5,35104E-06	0,021152187	2,98468E-12	1,20294E-12	2,89007E-06	0,042917635

Tabela 40 – Mediana dos resultados obtidos no cenário 4. Cada valor de parâmetro possui 30 instâncias e a mediana corresponde ao resultado do algoritmo nesses instâncias.

Cenário 4 - Avaliação de Escalabilidade												
r	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
100	92	24045	0,43	91	29586	3,52	91	29059,5	4,27	92	25232	1,28
200	186	48635	0,47	183	59290	6,96	183	58446	8,33	185	50972	2,52
300	280	73409	0,69	276	89290	10,41	276	87938,5	12,74	279	77053,5	3,78
400	374	97873,5	0,93	368	118927,5	13,82	369	117404,5	16,7	372	103076	5,05
500	467	122474,5	1,17	461	149369	17,52	461	146703,5	20,99	466	129322	6,3

d	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
100	43	18691,5	0,13	43	22136	4,94	43	21772	5,3	43	19382	1
200	41	28408	0,17	40	32515	28,4	40	31920,5	27,23	41	29694	1,8
300	39	36009,5	0,21	38	40182,5	109,47	38	39567	99,73	38	37668	3,29
400	37	42384	0,25	37	46417	315,38	37	45697	287,38	37	44391,5	5,39
500	36	47919,5	0,29	35	51814,5	733,62	35	51160	673,32	35	50281	7,71

n	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
4000	92	49575	0,21	90	61273,5	15,93	91	60465	19,51	92	51852,5	1,21
8000	94	64671	0,52	93	83856,5	30,28	93	82413,5	44,24	94	66956,5	2,46
12000	95	74966	0,96	94	99971	53,13	94	97997	85,45	95	76786	4,64
16000	95	84742	1,49	95	114694	84,68	95	113071,5	175,43	95	86528	7,62

s	MMVD			P-MCF			E-MCF			MMFOREST		
	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)	Z	Custo	Tempo(s)
10	45	11755,5	0,14	45	14592,5	1,75	45	14384	2,11	45	12316	0,82
20	45	11365,5	0,13	45	14027,5	2,19	45	14114	2,49	45	12198	1,28
30	45	11131	0,12	45	13248	2,68	45	13762,5	2,86	45	12037	1.879
40	46	10796	0,12	45	12792	3,26	45	13481,5	3,27	45	11838	2,42
50	46	10559,5	0,11	45	12586,5	3,94	45	12637,5	3,71	45	11742,5	3,01

Tabela 41 – Resultado do teste Post-hoc de Shaffer sobre as amostras obtidas pelos algoritmos no cenário 4 e parâmetros r, d, n e s . O nível de significância utilizado foi 0,05.

Cenário 4 - Avaliação de Escalabilidade									
r	Capacidade Residual			Custo			Tempo		
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
100	4,22487E-06	4,22487E-06	0,924865453	7,29711E-12	8,07156E-07	0,028611757	2,89007E-06	1,20294E-12	0,042917635
200	3,39752E-10	7,52943E-08	0,039928907	1,20294E-12	2,89007E-06	0,042917635	2,89007E-06	1,20294E-12	0,042917635
300	5,48478E-09	2,57931E-08	0,074672832	1,20294E-12	2,89007E-06	0,042917635	2,89007E-06	1,20294E-12	0,042917635
400	1,75795E-11	4,17368E-07	0,028611757	1,20294E-12	2,89007E-06	0,042917635	2,89007E-06	1,20294E-12	0,042917635
500	2,25017E-10	5,28763E-08	0,028611757	1,20294E-12	2,89007E-06	0,042917635	2,89007E-06	1,20294E-12	0,042917635
d	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
100	0,101572239	0,101572239	0,227258063	2,98468E-12	1,53845E-06	0,030337002	4,17368E-07	1,75795E-11	0,028611757
200	1,44836E-05	1,44836E-05	0,441342724	1,20294E-12	2,89007E-06	0,042917635	1,20294E-12	2,89007E-06	0,042917635
300	4,22487E-06	1,31206E-05	0,042917635	1,20294E-12	2,89007E-06	0,042917635	1,20294E-12	2,89007E-06	0,042917635
400	0,11978672	0,172948177	0,294745046	1,20294E-12	2,89007E-06	0,042917635	1,20294E-12	2,89007E-06	0,042917635
500	0,001431381	0,004352585	0,021152187	1,20294E-12	9,76526E-06	0,021152187	1,20294E-12	2,89007E-06	0,042917635
n	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
4000	3,02024E-07	4,12446E-05	0,540688281	2,98468E-12	1,53845E-06	0,030337002	2,89007E-06	1,20294E-12	0,042917635
8000	5,78014E-06	1,31206E-05	1	2,98468E-12	1,53845E-06	0,030337002	2,89007E-06	1,20294E-12	0,042917635
12000	7,24179E-06	1,31206E-05	1	1,20294E-12	2,89007E-06	0,042917635	2,89007E-06	1,20294E-12	0,042917635
16000	0,010737437	0,294745046	0,540688281	7,29711E-12	2,89007E-06	0,054972672	2,89007E-06	1,20294E-12	0,042917635
s	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests	P-MCF	E-MCF	MMForests
10	0,300261127	0,662014086	0,981560633	7,29711E-12	8,07156E-07	0,028611757	1,53845E-06	2,98468E-12	0,030337002
20	0,224018496	0,224018496	0,224018496	2,52337E-09	5,87413E-09	0,028611757	1,53845E-06	2,98468E-12	0,030337002
30	0,224018496	0,224018496	0,294745046	2,89007E-06	1,20294E-12	0,042917635	2,89007E-06	1,20294E-12	0,042917635
40	0,024000859	0,024000859	0,036144851	2,89007E-06	1,20294E-12	0,042917635	4,17368E-07	1,75795E-11	0,028611757
50	0,001815517	0,000686062	0,017571297	5,87413E-09	2,52337E-09	0,028611757	2,98468E-12	1,53845E-06	0,030337002

C Resultados do Modelo Multiobjetivo

Experimento 1

Neste apêndice, os resultados do experimento com instâncias cujo Pareto ótimo não foi determinado são apresentados nas páginas a seguir. Os algoritmos são comparados considerando as instâncias da Tabela 8.

A avaliação dos algoritmos seguiu a metodologia definida na Seção 7.6.3.1 e foi realizada considerando o teste de Friedman e o teste *PostHoc* de Shaffer. Este último teste foi aplicado diante da existência de diferença entre qualquer par de algoritmos comparados. Todos os testes estatísticos realizados utilizaram nível de significância igual a 0.05.

O resultado obtido da comparação dos algoritmos é apresentado por meio de mapas de calor. Quatro métricas foram consideradas na avaliação: I_H , $I_{\epsilon+}$, I_{ϵ} e tempo de processamento medido em segundos.

O resumo da avaliação dos resultados e conclusões sobre o desempenho dos algoritmos é apresentada na Seção 7.6.3.2.

C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 42 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $n = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.125	0.195	0.167	0.208	0.132	0.250	0.250	0.188	0.146	0.250	0.167	0.229	0.235
2	0.141	0.263	0.147	0.243	0.193	0.286	0.286	0.149	0.204	0.229	0.179	0.254	0.262
3	0.125	0.271	0.157	0.192	0.163	0.250	0.235	0.164	0.250	0.196	0.157	0.216	0.217
4	0.119	0.275	0.143	0.206	0.162	0.214	0.223	0.167	0.226	0.226	0.143	0.190	0.195
5	0.166	0.286	0.195	0.195	0.215	0.250	0.220	0.147	0.195	0.220	0.171	0.196	0.197
6	0.154	0.244	0.136	0.226	0.196	0.250	0.250	0.183	0.250	0.250	0.125	0.250	0.250
7	0.181	0.251	0.160	0.307	0.249	0.229	0.239	0.200	0.200	0.200	0.156	0.205	0.203
8	0.135	0.279	0.212	0.192	0.172	0.250	0.211	0.167	0.168	0.212	0.167	0.212	0.185
9	0.164	0.322	0.255	0.233	0.210	0.340	0.277	0.210	0.241	0.255	0.213	0.255	0.255
10	0.132	0.239	0.200	0.215	0.170	0.289	0.244	0.208	0.248	0.289	0.156	0.222	0.222
11	0.164	0.257	0.205	0.205	0.205	0.227	0.222	0.204	0.222	0.250	0.222	0.205	0.205
12	0.152	0.278	0.214	0.263	0.178	0.333	0.308	0.222	0.222	0.359	0.226	0.282	0.282
13	0.134	0.227	0.116	0.215	0.193	0.222	0.222	0.195	0.152	0.222	0.111	0.195	0.208
14	0.123	0.190	0.109	0.214	0.167	0.216	0.206	0.121	0.114	0.161	0.115	0.194	0.185
15	0.142	0.292	0.200	0.206	0.182	0.222	0.209	0.156	0.250	0.200	0.178	0.185	0.178
16	0.169	0.321	0.158	0.247	0.232	0.250	0.250	0.250	0.250	0.250	0.185	0.237	0.223
17	0.140	0.232	0.205	0.247	0.199	0.222	0.222	0.222	0.222	0.318	0.205	0.205	0.205
18	0.126	0.209	0.135	0.239	0.168	0.250	0.250	0.210	0.168	0.250	0.135	0.247	0.247
19	0.127	0.211	0.114	0.204	0.192	0.290	0.290	0.200	0.200	0.200	0.200	0.263	0.262
20	0.157	0.266	0.147	0.301	0.218	0.227	0.233	0.167	0.178	0.167	0.167	0.200	0.191

Figura 34 – *Posthoc* de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 20$.

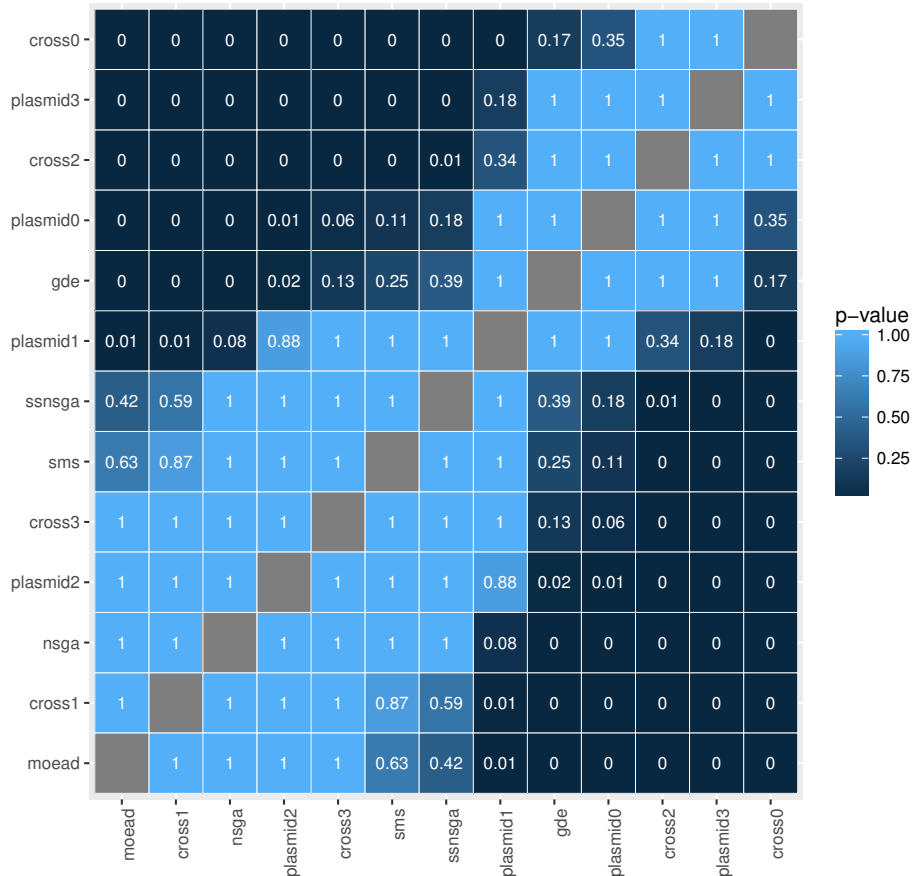
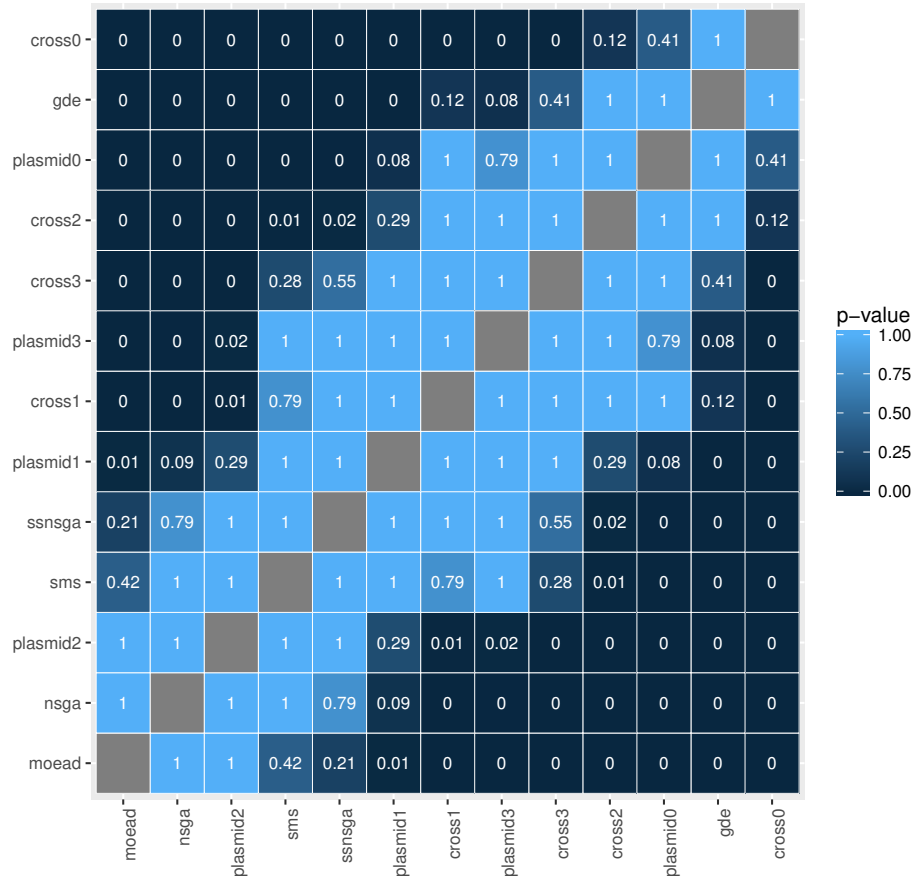


Tabela 43 – Medianas do indicador de qualidade I_H no conjunto com $n = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.123	0.204	0.305	0.324	0.202	0.390	0.345	0.274	0.274	0.414	0.326	0.306	0.304
2	0.123	0.250	0.217	0.341	0.221	0.434	0.422	0.259	0.333	0.405	0.344	0.376	0.363
3	0.120	0.337	0.180	0.269	0.199	0.354	0.334	0.268	0.366	0.330	0.300	0.301	0.305
4	0.125	0.290	0.213	0.237	0.200	0.335	0.334	0.256	0.375	0.350	0.313	0.284	0.288
5	0.196	0.386	0.293	0.321	0.261	0.375	0.360	0.257	0.339	0.343	0.330	0.325	0.327
6	0.159	0.320	0.246	0.224	0.229	0.400	0.401	0.246	0.376	0.356	0.215	0.344	0.328
7	0.225	0.295	0.298	0.382	0.288	0.448	0.447	0.241	0.310	0.342	0.311	0.409	0.402
8	0.186	0.325	0.291	0.273	0.242	0.340	0.323	0.283	0.304	0.382	0.304	0.304	0.302
9	0.236	0.367	0.323	0.268	0.292	0.476	0.435	0.321	0.381	0.383	0.316	0.404	0.395
10	0.147	0.209	0.327	0.224	0.207	0.420	0.346	0.294	0.302	0.496	0.279	0.296	0.304
11	0.180	0.305	0.305	0.296	0.279	0.385	0.358	0.264	0.328	0.392	0.329	0.326	0.335
12	0.203	0.342	0.281	0.376	0.269	0.495	0.456	0.317	0.367	0.521	0.417	0.420	0.416
13	0.146	0.255	0.185	0.256	0.235	0.332	0.345	0.223	0.272	0.322	0.237	0.289	0.289
14	0.101	0.161	0.170	0.226	0.161	0.302	0.281	0.163	0.203	0.262	0.196	0.261	0.245
15	0.225	0.358	0.333	0.306	0.284	0.381	0.369	0.240	0.370	0.312	0.303	0.339	0.333
16	0.245	0.398	0.333	0.350	0.263	0.447	0.431	0.348	0.397	0.425	0.346	0.402	0.390
17	0.206	0.302	0.344	0.325	0.270	0.361	0.350	0.303	0.328	0.548	0.326	0.314	0.299
18	0.164	0.314	0.311	0.295	0.230	0.396	0.393	0.289	0.280	0.446	0.303	0.346	0.337
19	0.126	0.281	0.222	0.211	0.223	0.477	0.463	0.241	0.274	0.342	0.296	0.427	0.421
20	0.218	0.340	0.264	0.368	0.305	0.401	0.408	0.253	0.314	0.325	0.335	0.366	0.347

Figura 35 – *Posthoc* de Shaffer para I_H no conjunto com $n = 20$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 44 – Medianas do indicador de qualidade I_e no conjunto com $n = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.112	1.195	1.167	1.208	1.132	1.250	1.235	1.163	1.140	1.250	1.167	1.216	1.222
2	1.141	1.263	1.147	1.243	1.193	1.265	1.249	1.145	1.182	1.221	1.162	1.224	1.222
3	1.125	1.271	1.157	1.192	1.163	1.231	1.208	1.154	1.250	1.196	1.157	1.208	1.204
4	1.118	1.275	1.143	1.206	1.162	1.203	1.201	1.148	1.223	1.226	1.143	1.172	1.176
5	1.166	1.286	1.195	1.195	1.215	1.244	1.210	1.140	1.189	1.220	1.171	1.195	1.185
6	1.154	1.242	1.136	1.226	1.196	1.250	1.250	1.151	1.250	1.250	1.125	1.227	1.218
7	1.181	1.251	1.160	1.307	1.249	1.223	1.234	1.200	1.200	1.200	1.154	1.200	1.200
8	1.135	1.279	1.212	1.192	1.172	1.231	1.181	1.167	1.168	1.212	1.167	1.182	1.171
9	1.163	1.322	1.255	1.233	1.210	1.308	1.235	1.200	1.205	1.255	1.206	1.219	1.212
10	1.132	1.239	1.200	1.215	1.170	1.275	1.213	1.181	1.222	1.281	1.156	1.199	1.196
11	1.164	1.257	1.205	1.205	1.205	1.225	1.200	1.180	1.222	1.250	1.213	1.196	1.188
12	1.152	1.278	1.205	1.263	1.178	1.300	1.271	1.222	1.222	1.359	1.226	1.236	1.241
13	1.134	1.227	1.116	1.215	1.193	1.222	1.222	1.171	1.140	1.222	1.111	1.190	1.193
14	1.123	1.190	1.109	1.214	1.167	1.194	1.194	1.111	1.114	1.161	1.111	1.178	1.165
15	1.142	1.292	1.200	1.206	1.182	1.217	1.191	1.140	1.250	1.200	1.178	1.172	1.159
16	1.169	1.321	1.158	1.247	1.232	1.250	1.248	1.250	1.250	1.250	1.184	1.215	1.207
17	1.140	1.232	1.205	1.247	1.199	1.213	1.209	1.200	1.222	1.318	1.200	1.191	1.200
18	1.126	1.209	1.135	1.239	1.168	1.250	1.250	1.190	1.154	1.250	1.132	1.231	1.221
19	1.127	1.211	1.114	1.204	1.192	1.266	1.259	1.200	1.200	1.200	1.200	1.237	1.238
20	1.157	1.266	1.147	1.301	1.218	1.197	1.214	1.167	1.167	1.167	1.167	1.187	1.179

Figura 36 – *Posthoc* de Shaffer para I_e no conjunto com $n = 20$.

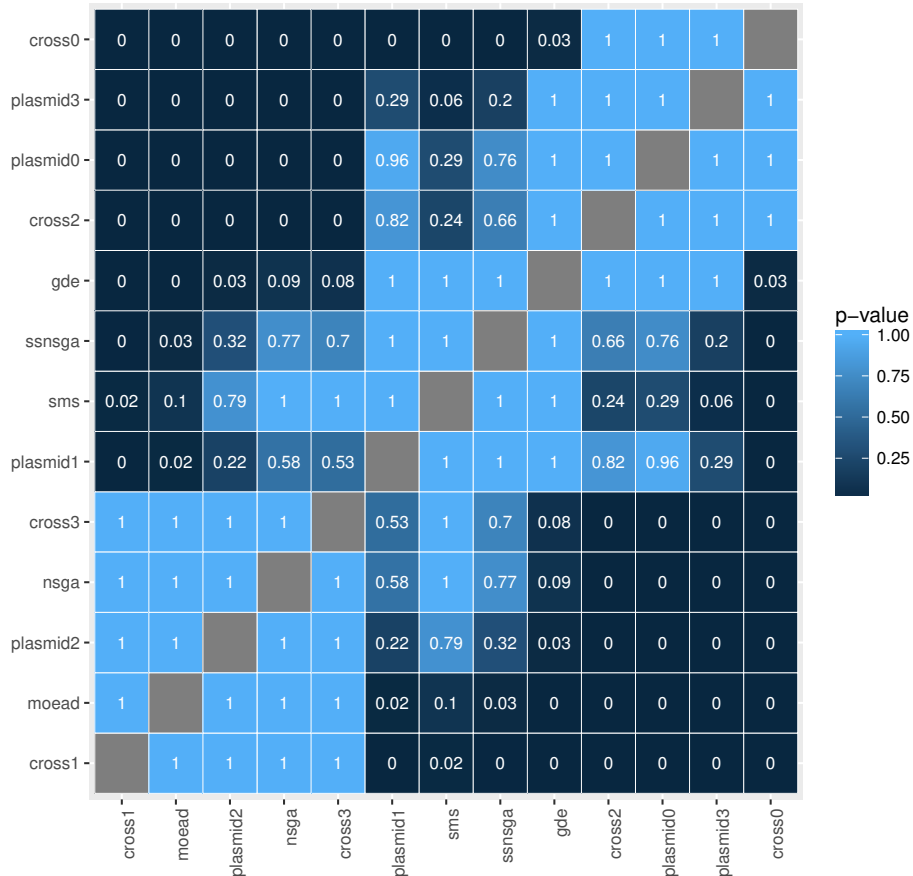
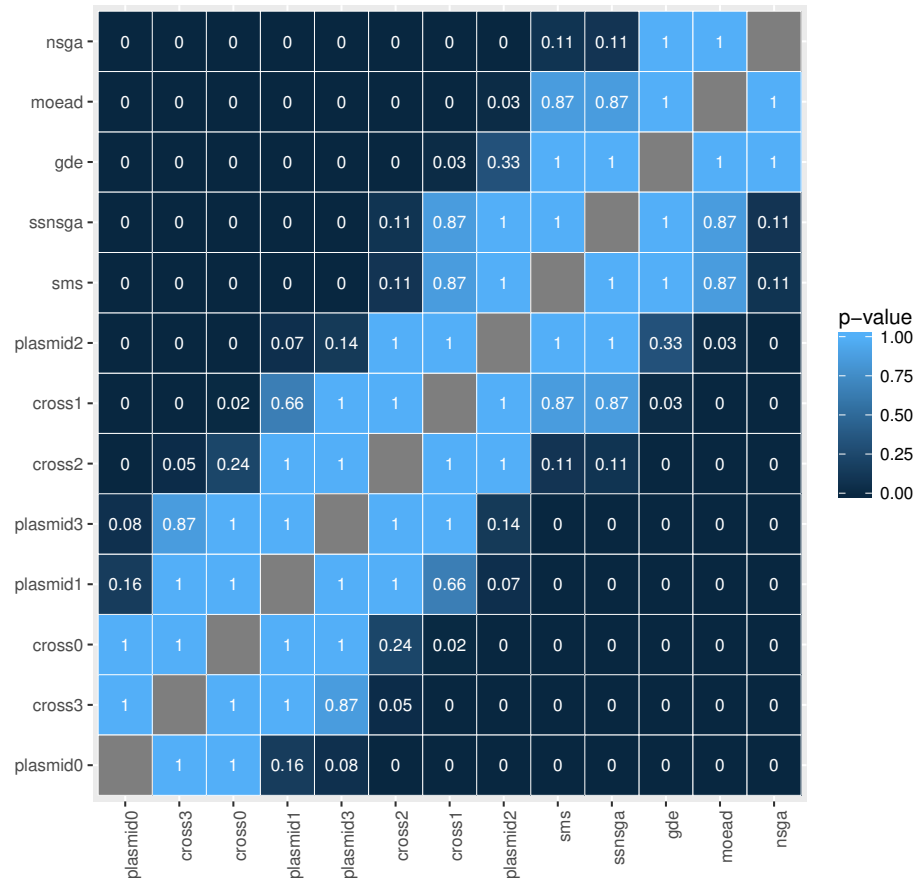


Tabela 45 – Medianas do tempo de processamento no conjunto com $n = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.985	1.407	1.632	2.031	0.711	0.408	0.171	2.216	1.856	1.102	1.839	1.009	1.008
2	2.000	1.420	1.648	2.068	0.722	0.398	0.165	2.250	1.865	1.133	1.897	1.037	1.037
3	1.903	1.413	1.530	1.934	0.657	0.367	0.159	2.113	1.799	1.043	1.693	1.013	0.972
4	1.819	1.304	1.514	1.885	0.634	0.381	0.156	2.050	1.691	1.032	1.681	0.945	0.940
5	1.987	1.421	1.628	2.073	0.680	0.389	0.164	2.192	1.834	1.094	1.805	1.013	1.012
6	2.174	1.549	1.782	2.224	0.781	0.438	0.189	2.422	2.068	1.219	2.008	1.090	1.093
7	1.895	1.364	1.580	1.955	0.700	0.381	0.160	2.120	1.762	1.052	1.730	0.962	0.960
8	2.071	1.455	1.719	2.141	0.684	0.423	0.179	2.341	1.899	1.191	1.926	1.038	1.040
9	2.226	1.617	1.800	2.248	0.764	0.440	0.188	2.485	2.093	1.227	2.052	1.126	1.122
10	1.988	1.415	1.636	2.024	0.684	0.400	0.171	2.232	1.861	1.132	1.822	1.006	1.007
11	1.934	1.409	1.573	2.003	0.663	0.386	0.166	2.160	1.814	1.081	1.779	0.999	0.993
12	2.303	1.640	1.885	2.382	0.816	0.463	0.199	2.572	2.157	1.266	2.121	1.156	1.154
13	2.016	1.446	1.668	2.055	0.746	0.403	0.171	2.274	1.900	1.139	1.879	1.030	1.033
14	1.732	1.253	1.425	1.808	0.656	0.359	0.148	1.946	1.608	0.967	1.623	0.910	0.910
15	1.854	1.330	1.520	1.965	0.739	0.368	0.152	2.060	1.717	1.029	1.716	0.954	0.953
16	2.045	1.475	1.678	2.107	0.751	0.409	0.174	2.282	1.899	1.144	1.874	1.045	1.043
17	1.900	1.369	1.553	1.927	0.697	0.374	0.161	2.121	1.755	1.068	1.726	0.966	0.959
18	2.011	1.437	1.653	2.049	0.720	0.397	0.165	2.266	1.871	1.140	1.830	1.033	1.035
19	2.081	1.490	1.694	2.189	0.754	0.424	0.177	2.329	1.953	1.174	1.964	1.064	1.062
20	1.947	1.410	1.590	1.993	0.715	0.393	0.169	2.146	1.810	1.069	1.764	0.990	0.991

Figura 37 – *Posthoc* de Shaffer para o Tempo no conjunto com $n = 20$.



C. RESULTADOS DO MODELO MULTIOBJETIVO

EXPERIMENTO 1

Tabela 46 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $n = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsa	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsa
1	0.078	0.149	0.189	0.189	0.140	0.231	0.189	0.154	0.231	0.216	0.243	0.189	0.162
2	0.167	0.167	0.071	0.137	0.167	0.237	0.222	0.194	0.214	0.167	0.143	0.214	0.218
3	0.118	0.171	0.118	0.141	0.176	0.273	0.248	0.182	0.182	0.176	0.118	0.235	0.213
4	0.079	0.147	0.206	0.118	0.118	0.203	0.176	0.133	0.140	0.235	0.147	0.176	0.147
5	0.083	0.152	0.083	0.155	0.116	0.250	0.195	0.113	0.172	0.167	0.089	0.178	0.178
6	0.065	0.101	0.258	0.226	0.104	0.226	0.188	0.129	0.129	0.290	0.226	0.188	0.188
7	0.077	0.172	0.216	0.162	0.118	0.216	0.162	0.154	0.154	0.189	0.162	0.154	0.154
8	0.079	0.129	0.257	0.229	0.114	0.171	0.158	0.114	0.158	0.343	0.257	0.143	0.145
9	0.083	0.144	0.286	0.200	0.114	0.257	0.200	0.171	0.250	0.286	0.200	0.200	0.200
10	0.096	0.181	0.167	0.181	0.147	0.167	0.167	0.167	0.250	0.250	0.167	0.167	0.167
11	0.091	0.233	0.091	0.166	0.155	0.214	0.214	0.182	0.182	0.182	0.146	0.182	0.184
12	0.100	0.198	0.277	0.213	0.133	0.213	0.190	0.170	0.133	0.298	0.234	0.170	0.170
13	0.083	0.132	0.194	0.122	0.139	0.231	0.194	0.154	0.167	0.194	0.136	0.194	0.194
14	0.083	0.167	0.200	0.167	0.150	0.250	0.250	0.167	0.250	0.250	0.200	0.242	0.217
15	0.105	0.163	0.190	0.214	0.183	0.190	0.190	0.158	0.211	0.214	0.214	0.158	0.158
16	0.077	0.137	0.154	0.128	0.129	0.179	0.167	0.154	0.128	0.179	0.167	0.167	0.167
17	0.063	0.124	0.114	0.133	0.122	0.200	0.188	0.125	0.125	0.171	0.125	0.157	0.171
18	0.071	0.203	0.259	0.168	0.123	0.204	0.153	0.143	0.214	0.222	0.143	0.143	0.143
19	0.083	0.167	0.189	0.158	0.130	0.216	0.189	0.167	0.333	0.243	0.178	0.167	0.167
20	0.077	0.162	0.109	0.128	0.127	0.224	0.193	0.154	0.154	0.183	0.154	0.177	0.174

Figura 38 – *Posthoc* de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 40$.

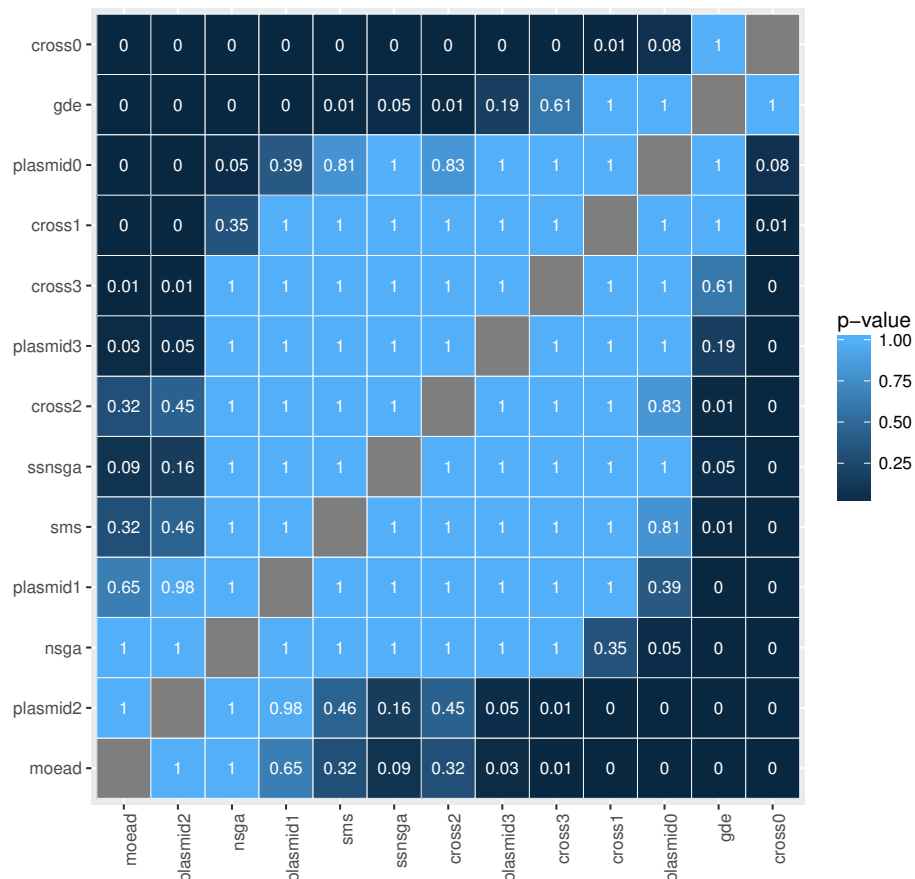
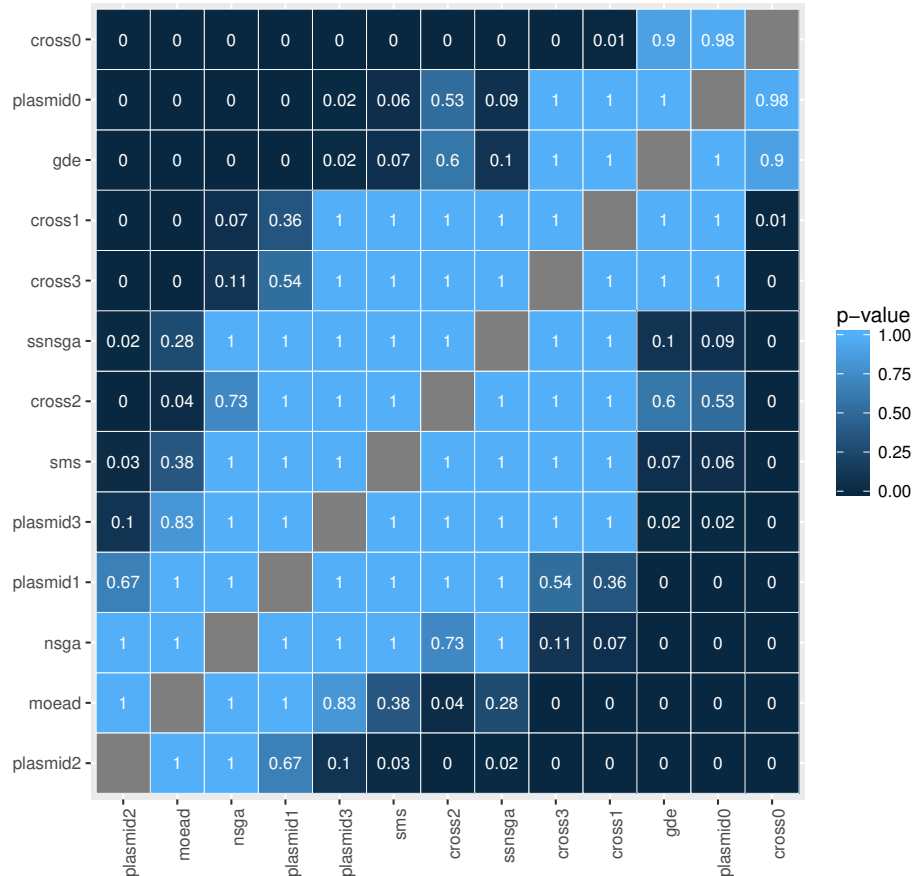


Tabela 47 – Medianas do indicador de qualidade I_H no conjunto com $n = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.100	0.229	0.221	0.277	0.218	0.290	0.271	0.191	0.335	0.317	0.346	0.257	0.251
2	0.279	0.324	0.159	0.234	0.334	0.423	0.401	0.346	0.414	0.321	0.268	0.365	0.364
3	0.196	0.367	0.245	0.159	0.270	0.398	0.394	0.285	0.387	0.268	0.177	0.377	0.372
4	0.147	0.223	0.289	0.191	0.200	0.299	0.272	0.211	0.281	0.380	0.295	0.261	0.254
5	0.147	0.280	0.153	0.203	0.175	0.310	0.303	0.152	0.309	0.266	0.175	0.279	0.279
6	0.139	0.178	0.313	0.282	0.182	0.329	0.306	0.191	0.225	0.409	0.301	0.292	0.296
7	0.139	0.288	0.297	0.270	0.209	0.306	0.289	0.226	0.245	0.331	0.335	0.281	0.273
8	0.121	0.208	0.314	0.338	0.196	0.256	0.233	0.170	0.237	0.438	0.337	0.210	0.205
9	0.158	0.233	0.335	0.248	0.207	0.360	0.323	0.264	0.369	0.410	0.317	0.292	0.304
10	0.179	0.311	0.270	0.314	0.247	0.337	0.333	0.229	0.336	0.379	0.300	0.314	0.310
11	0.072	0.249	0.137	0.159	0.172	0.383	0.377	0.233	0.293	0.316	0.200	0.344	0.328
12	0.162	0.246	0.334	0.321	0.256	0.330	0.279	0.241	0.246	0.401	0.312	0.258	0.253
13	0.092	0.213	0.242	0.216	0.187	0.291	0.284	0.179	0.255	0.324	0.207	0.267	0.263
14	0.173	0.287	0.358	0.325	0.271	0.378	0.372	0.281	0.371	0.434	0.369	0.347	0.346
15	0.176	0.261	0.271	0.305	0.293	0.331	0.314	0.219	0.289	0.349	0.335	0.295	0.297
16	0.080	0.147	0.209	0.228	0.189	0.257	0.258	0.180	0.199	0.332	0.291	0.190	0.193
17	0.115	0.202	0.211	0.245	0.239	0.248	0.241	0.157	0.203	0.282	0.225	0.224	0.219
18	0.144	0.316	0.354	0.260	0.206	0.308	0.296	0.235	0.316	0.404	0.272	0.248	0.239
19	0.112	0.237	0.267	0.248	0.196	0.292	0.281	0.238	0.452	0.402	0.299	0.232	0.265
20	0.130	0.227	0.200	0.174	0.211	0.302	0.262	0.236	0.244	0.341	0.263	0.250	0.235

Figura 39 – *Posthoc* de Shaffer para I_H no conjunto com $n = 40$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 48 – Medianas do indicador de qualidade I_e no conjunto com $n = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.078	1.149	1.189	1.189	1.137	1.214	1.179	1.154	1.231	1.216	1.243	1.179	1.154
2	1.167	1.167	1.071	1.137	1.167	1.222	1.213	1.189	1.214	1.160	1.143	1.200	1.200
3	1.118	1.171	1.118	1.141	1.176	1.273	1.235	1.176	1.182	1.176	1.115	1.235	1.206
4	1.079	1.147	1.206	1.118	1.118	1.200	1.169	1.133	1.136	1.235	1.147	1.176	1.139
5	1.083	1.151	1.083	1.154	1.116	1.250	1.180	1.108	1.172	1.167	1.084	1.178	1.178
6	1.065	1.101	1.258	1.226	1.104	1.212	1.176	1.125	1.125	1.290	1.226	1.176	1.176
7	1.077	1.172	1.216	1.162	1.117	1.205	1.154	1.154	1.154	1.189	1.162	1.154	1.154
8	1.072	1.129	1.257	1.229	1.109	1.158	1.138	1.110	1.158	1.343	1.257	1.130	1.134
9	1.083	1.144	1.286	1.200	1.114	1.244	1.184	1.167	1.250	1.286	1.200	1.184	1.171
10	1.096	1.181	1.167	1.181	1.147	1.167	1.167	1.167	1.250	1.250	1.167	1.167	1.167
11	1.085	1.233	1.091	1.166	1.155	1.193	1.189	1.182	1.182	1.182	1.131	1.182	1.182
12	1.098	1.198	1.277	1.213	1.132	1.194	1.163	1.143	1.133	1.298	1.234	1.148	1.139
13	1.077	1.132	1.194	1.122	1.135	1.221	1.189	1.154	1.162	1.194	1.125	1.182	1.170
14	1.083	1.167	1.200	1.167	1.133	1.235	1.231	1.167	1.250	1.250	1.200	1.228	1.194
15	1.105	1.163	1.190	1.214	1.183	1.174	1.165	1.158	1.211	1.214	1.214	1.158	1.158
16	1.077	1.137	1.154	1.128	1.129	1.179	1.167	1.150	1.127	1.179	1.167	1.165	1.167
17	1.063	1.122	1.114	1.132	1.122	1.188	1.175	1.125	1.125	1.169	1.125	1.148	1.154
18	1.071	1.203	1.259	1.168	1.123	1.193	1.143	1.143	1.214	1.222	1.143	1.143	1.143
19	1.083	1.167	1.189	1.158	1.125	1.205	1.179	1.167	1.333	1.237	1.178	1.167	1.167
20	1.077	1.162	1.109	1.128	1.127	1.214	1.172	1.154	1.154	1.174	1.154	1.171	1.164

Figura 40 – *Posthoc* de Shaffer para I_e no conjunto com $n = 40$.

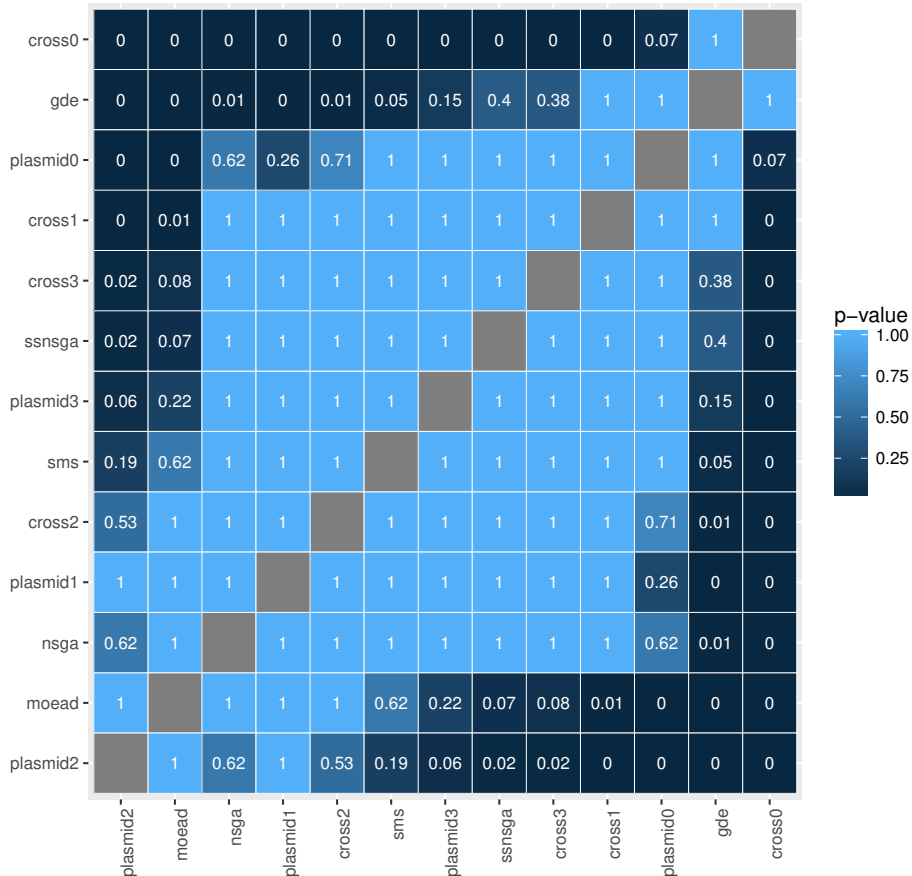
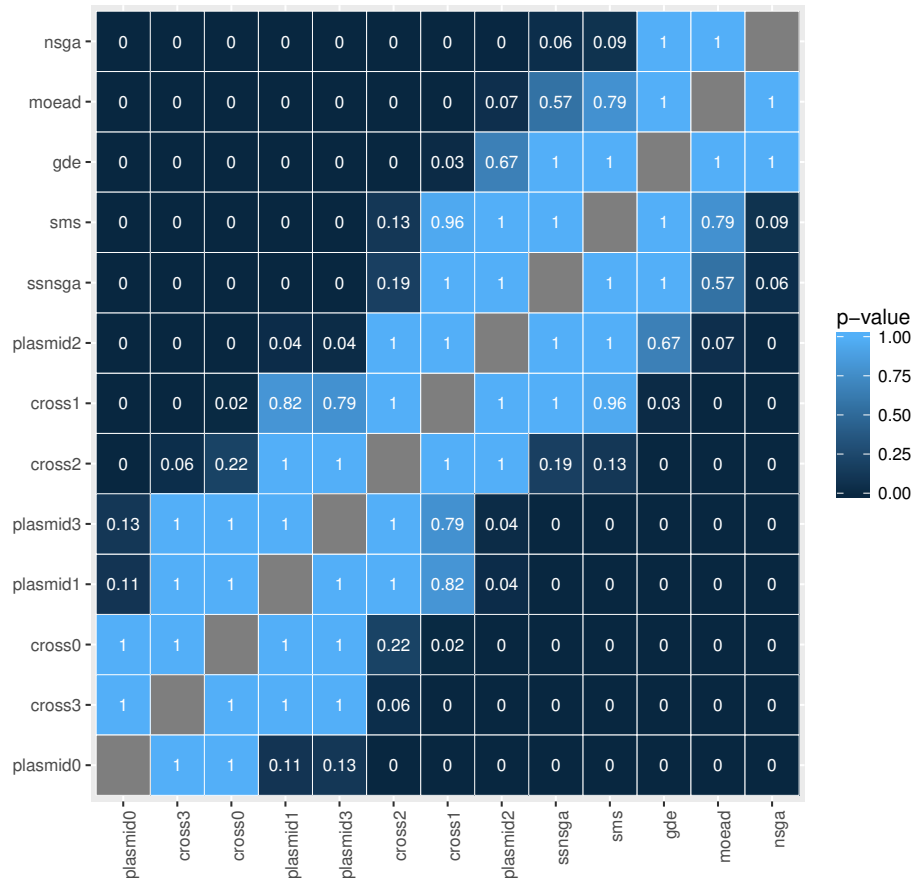


Tabela 49 – Medianas do tempo de processamento no conjunto com $n = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	2.041	1.460	1.678	2.069	0.699	0.513	0.210	2.260	1.872	1.143	1.881	1.119	1.127
2	2.194	1.558	1.808	2.194	0.798	0.551	0.223	2.443	2.022	1.231	2.010	1.189	1.197
3	1.955	1.387	1.629	2.026	0.721	0.510	0.205	2.161	1.780	1.110	1.820	1.064	1.065
4	2.151	1.513	1.783	2.182	0.714	0.558	0.231	2.401	1.989	1.228	2.004	1.177	1.178
5	1.979	1.388	1.666	2.009	0.708	0.530	0.215	2.253	1.794	1.160	1.842	1.084	1.089
6	2.105	1.494	1.757	2.173	0.720	0.547	0.217	2.368	1.970	1.178	1.940	1.150	1.151
7	2.140	1.515	1.765	2.185	0.737	0.538	0.222	2.349	1.992	1.189	1.953	1.163	1.167
8	1.843	1.339	1.495	1.877	0.610	0.464	0.185	2.046	1.725	1.027	1.686	1.023	1.026
9	2.137	1.521	1.773	2.134	0.734	0.526	0.216	2.421	1.972	1.219	2.039	1.146	1.154
10	2.079	1.478	1.711	2.118	0.755	0.509	0.206	2.303	1.929	1.155	1.920	1.143	1.147
11	2.141	1.518	1.797	2.175	0.792	0.563	0.228	2.413	1.990	1.246	1.987	1.150	1.148
12	2.261	1.619	1.880	2.275	0.781	0.569	0.240	2.551	2.119	1.284	2.089	1.227	1.233
13	2.127	1.528	1.723	2.170	0.755	0.503	0.205	2.330	1.949	1.149	1.928	1.156	1.171
14	2.103	1.508	1.723	2.162	0.731	0.534	0.218	2.363	1.943	1.162	1.998	1.166	1.169
15	2.532	1.824	2.109	2.536	0.878	0.618	0.282	2.902	2.328	1.398	2.458	1.350	1.367
16	2.062	1.454	1.728	2.159	0.728	0.527	0.213	2.329	1.925	1.151	1.949	1.140	1.146
17	2.130	1.546	1.752	2.162	0.799	0.535	0.218	2.371	2.006	1.168	1.956	1.191	1.192
18	2.263	1.578	1.909	2.269	0.791	0.562	0.226	2.559	2.101	1.313	2.109	1.223	1.226
19	2.082	1.490	1.718	2.096	0.775	0.522	0.213	2.329	1.932	1.183	1.908	1.132	1.142
20	2.191	1.541	1.817	2.244	0.783	0.556	0.226	2.505	2.038	1.262	2.092	1.193	1.195

Figura 41 – *Posthoc* de Shaffer para o Tempo no conjunto com $n = 40$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 50 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $n = 60$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.064	0.113	0.050	0.101	0.125	0.200	0.175	0.123	0.132	0.150	0.100	0.175	0.168
2	0.105	0.165	0.184	0.158	0.157	0.184	0.167	0.157	0.158	0.184	0.184	0.167	0.167
3	0.073	0.131	0.220	0.195	0.165	0.195	0.171	0.127	0.125	0.244	0.195	0.154	0.143
4	0.057	0.109	0.063	0.095	0.086	0.188	0.148	0.114	0.125	0.125	0.063	0.143	0.129
5	0.089	0.143	0.244	0.267	0.119	0.200	0.167	0.111	0.133	0.244	0.267	0.167	0.167
6	0.061	0.115	0.061	0.109	0.117	0.158	0.152	0.129	0.158	0.158	0.105	0.152	0.152
7	0.087	0.173	0.207	0.130	0.148	0.217	0.174	0.167	0.222	0.217	0.159	0.167	0.168
8	0.076	0.174	0.125	0.168	0.131	0.214	0.164	0.133	0.133	0.143	0.121	0.161	0.145
9	0.071	0.136	0.143	0.143	0.152	0.214	0.214	0.143	0.214	0.214	0.143	0.188	0.188
10	0.081	0.183	0.164	0.145	0.138	0.183	0.164	0.167	0.154	0.182	0.145	0.164	0.164
11	0.081	0.133	0.216	0.216	0.129	0.243	0.200	0.133	0.195	0.243	0.195	0.189	0.189
12	0.063	0.111	0.281	0.250	0.125	0.222	0.188	0.119	0.161	0.250	0.250	0.177	0.167
13	0.088	0.161	0.195	0.155	0.176	0.158	0.158	0.121	0.158	0.195	0.149	0.147	0.146
14	0.111	0.210	0.222	0.222	0.133	0.256	0.200	0.200	0.200	0.267	0.222	0.200	0.200
15	0.063	0.110	0.063	0.121	0.104	0.188	0.171	0.125	0.125	0.125	0.110	0.146	0.146
16	0.063	0.115	0.095	0.128	0.133	0.156	0.143	0.101	0.143	0.143	0.095	0.133	0.143
17	0.105	0.134	0.220	0.317	0.139	0.268	0.213	0.158	0.211	0.488	0.232	0.211	0.211
18	0.071	0.163	0.232	0.161	0.113	0.214	0.167	0.161	0.167	0.250	0.196	0.167	0.167
19	0.079	0.182	0.149	0.151	0.141	0.234	0.214	0.213	0.214	0.214	0.173	0.214	0.214
20	0.085	0.143	0.149	0.128	0.113	0.234	0.200	0.155	0.200	0.200	0.133	0.200	0.200

Figura 42 – *Posthoc* de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 60$.

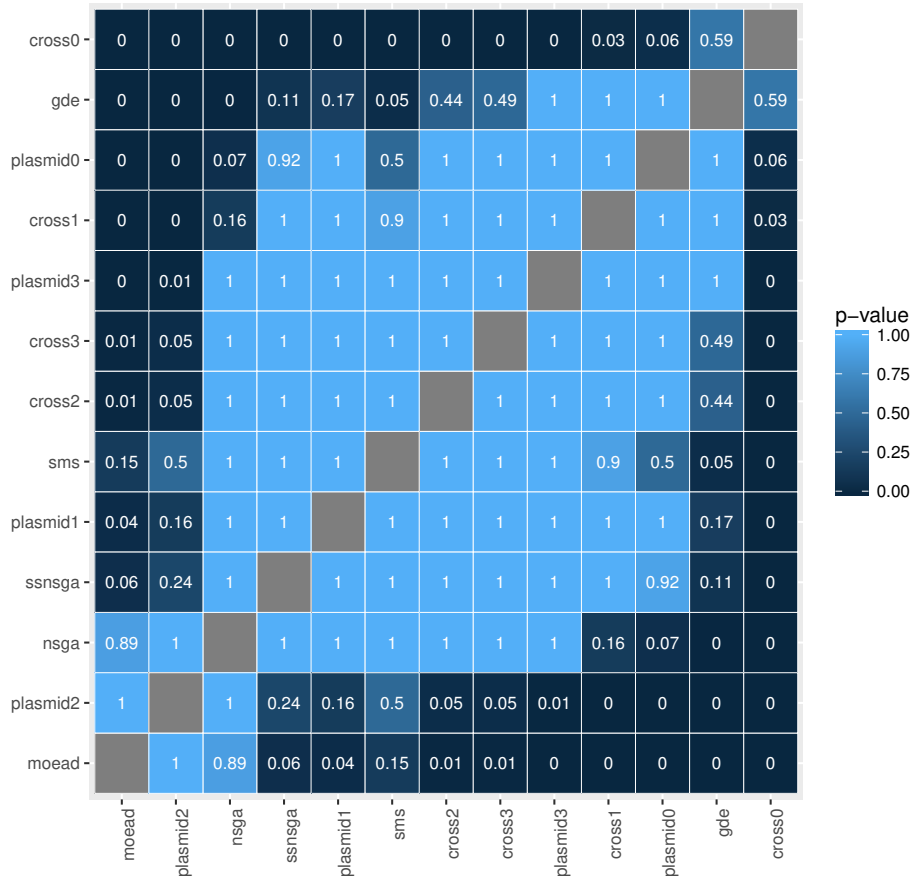
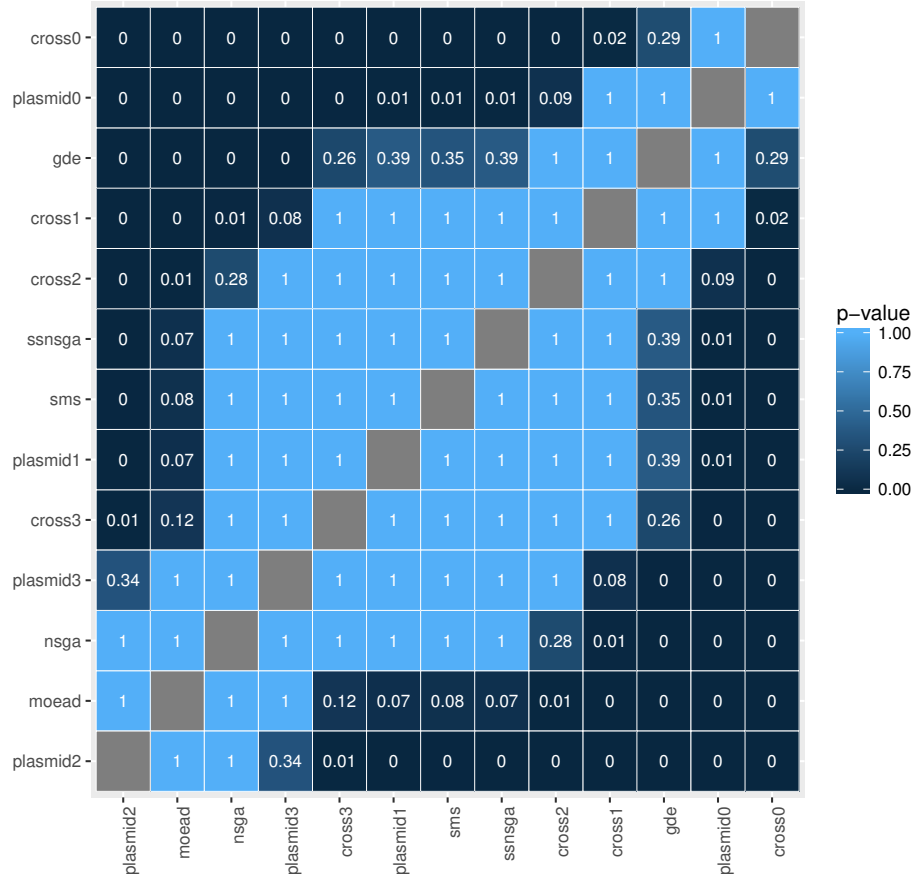


Tabela 51 – Medianas do indicador de qualidade I_H no conjunto com $n = 60$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.076	0.180	0.085	0.127	0.175	0.255	0.241	0.151	0.183	0.181	0.164	0.225	0.223
2	0.144	0.252	0.275	0.286	0.249	0.277	0.267	0.194	0.225	0.336	0.319	0.251	0.257
3	0.103	0.201	0.273	0.301	0.243	0.246	0.242	0.142	0.198	0.322	0.283	0.226	0.217
4	0.062	0.131	0.110	0.133	0.135	0.230	0.224	0.111	0.196	0.195	0.141	0.198	0.200
5	0.134	0.229	0.274	0.322	0.224	0.267	0.256	0.187	0.216	0.325	0.341	0.237	0.224
6	0.134	0.207	0.151	0.178	0.202	0.236	0.252	0.180	0.261	0.293	0.224	0.221	0.219
7	0.165	0.282	0.287	0.254	0.257	0.348	0.314	0.244	0.314	0.371	0.280	0.292	0.309
8	0.098	0.205	0.222	0.245	0.193	0.279	0.258	0.178	0.245	0.278	0.231	0.235	0.224
9	0.177	0.218	0.235	0.267	0.255	0.349	0.341	0.259	0.323	0.370	0.300	0.318	0.315
10	0.138	0.262	0.259	0.249	0.212	0.281	0.252	0.194	0.228	0.342	0.271	0.234	0.237
11	0.122	0.229	0.287	0.273	0.206	0.304	0.275	0.200	0.233	0.374	0.331	0.266	0.263
12	0.128	0.226	0.332	0.300	0.211	0.304	0.279	0.204	0.217	0.365	0.347	0.252	0.259
13	0.146	0.248	0.278	0.269	0.259	0.282	0.286	0.189	0.247	0.324	0.293	0.263	0.256
14	0.159	0.330	0.314	0.353	0.242	0.360	0.330	0.245	0.319	0.409	0.337	0.306	0.312
15	0.144	0.222	0.144	0.172	0.210	0.250	0.238	0.178	0.235	0.215	0.164	0.213	0.211
16	0.126	0.203	0.190	0.263	0.224	0.247	0.249	0.168	0.245	0.263	0.236	0.228	0.224
17	0.148	0.219	0.324	0.452	0.245	0.409	0.335	0.217	0.279	0.633	0.349	0.307	0.301
18	0.122	0.234	0.306	0.258	0.197	0.309	0.265	0.184	0.263	0.356	0.304	0.251	0.256
19	0.146	0.294	0.230	0.267	0.232	0.363	0.351	0.277	0.331	0.394	0.331	0.311	0.309
20	0.157	0.279	0.266	0.241	0.225	0.380	0.335	0.221	0.330	0.337	0.283	0.315	0.321

Figura 43 – *Posthoc* de Shaffer para I_H no conjunto com $n = 60$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 52 – Medianas do indicador de qualidade I_ϵ no conjunto com $n = 60$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.062	1.113	1.050	1.101	1.123	1.200	1.175	1.116	1.125	1.150	1.100	1.175	1.160
2	1.100	1.165	1.184	1.158	1.156	1.179	1.167	1.138	1.143	1.184	1.184	1.154	1.159
3	1.071	1.131	1.220	1.195	1.165	1.190	1.156	1.117	1.116	1.244	1.195	1.146	1.143
4	1.057	1.108	1.063	1.095	1.085	1.188	1.143	1.114	1.125	1.125	1.063	1.143	1.125
5	1.082	1.143	1.244	1.267	1.118	1.184	1.158	1.111	1.122	1.244	1.267	1.150	1.150
6	1.057	1.115	1.061	1.109	1.114	1.158	1.147	1.120	1.158	1.158	1.105	1.145	1.143
7	1.084	1.173	1.207	1.130	1.148	1.200	1.167	1.167	1.222	1.217	1.144	1.167	1.167
8	1.075	1.174	1.125	1.168	1.131	1.207	1.158	1.133	1.133	1.143	1.109	1.155	1.138
9	1.071	1.136	1.125	1.143	1.152	1.206	1.197	1.143	1.214	1.207	1.143	1.185	1.164
10	1.080	1.183	1.164	1.145	1.138	1.172	1.150	1.159	1.138	1.182	1.145	1.150	1.150
11	1.079	1.133	1.216	1.216	1.108	1.227	1.182	1.133	1.179	1.243	1.189	1.176	1.159
12	1.063	1.111	1.281	1.250	1.118	1.200	1.169	1.113	1.155	1.250	1.250	1.158	1.152
13	1.088	1.161	1.195	1.155	1.176	1.158	1.158	1.107	1.158	1.195	1.146	1.143	1.130
14	1.105	1.210	1.222	1.222	1.133	1.221	1.200	1.164	1.200	1.256	1.222	1.200	1.200
15	1.063	1.110	1.063	1.120	1.102	1.188	1.167	1.125	1.125	1.125	1.107	1.146	1.146
16	1.063	1.115	1.095	1.128	1.133	1.153	1.143	1.095	1.143	1.143	1.095	1.126	1.136
17	1.098	1.134	1.220	1.317	1.133	1.263	1.211	1.150	1.211	1.488	1.221	1.204	1.207
18	1.067	1.163	1.232	1.161	1.113	1.190	1.167	1.141	1.167	1.250	1.196	1.167	1.167
19	1.078	1.182	1.149	1.151	1.141	1.214	1.214	1.172	1.214	1.214	1.160	1.196	1.214
20	1.082	1.143	1.149	1.128	1.109	1.222	1.200	1.143	1.200	1.200	1.133	1.200	1.200

Figura 44 – *Posthoc* de Shaffer para I_ϵ no conjunto com $n = 60$.

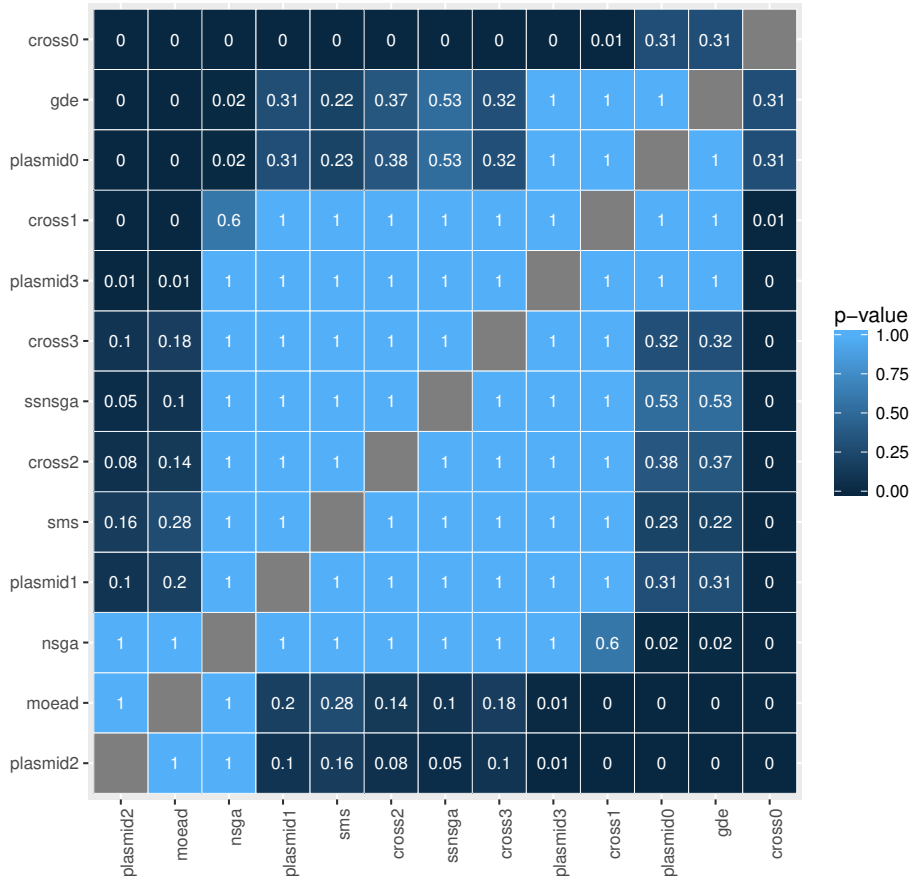
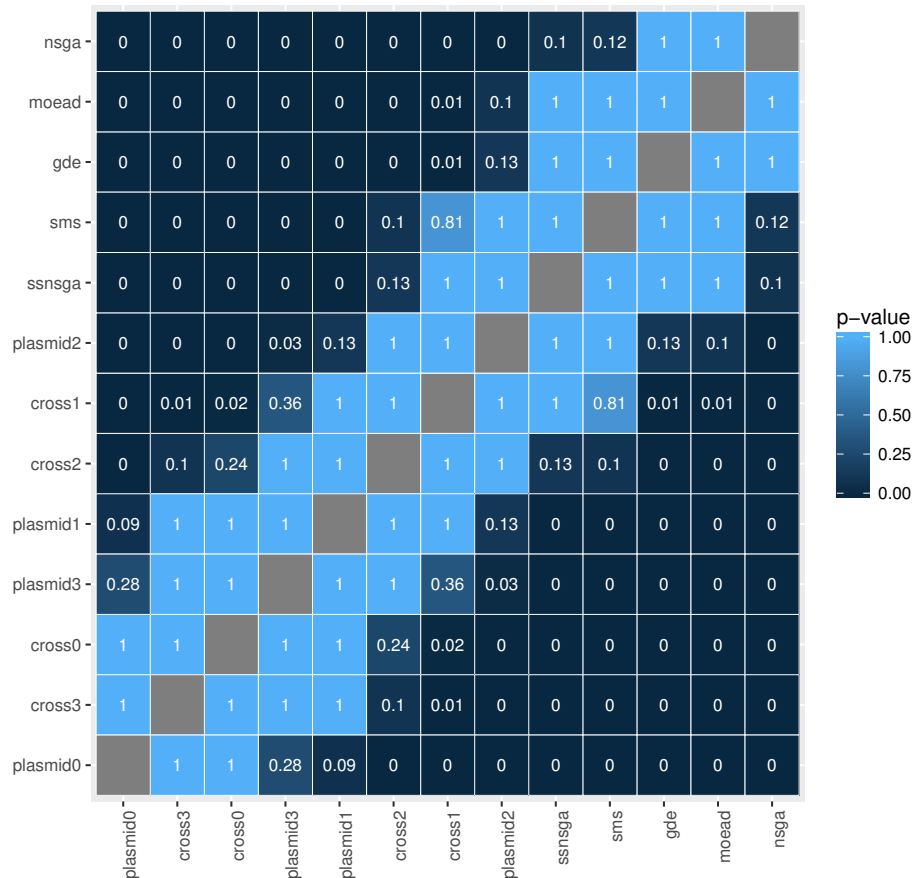


Tabela 53 – Medianas do tempo de processamento no conjunto com $n = 60$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	2.322	1.665	1.950	2.368	0.789	0.714	0.289	2.604	2.171	1.338	2.201	1.240	1.246
2	2.428	1.755	2.022	2.504	0.767	0.753	0.321	2.797	2.316	1.397	2.352	1.281	1.294
3	2.673	1.928	2.222	2.678	0.821	0.806	0.350	3.065	2.550	1.490	2.561	1.405	1.411
4	2.062	1.463	1.740	2.097	0.729	0.639	0.249	2.340	1.894	1.192	1.928	1.108	1.107
5	2.188	1.607	1.765	2.204	0.678	0.685	0.291	2.472	2.104	1.244	2.066	1.185	1.193
6	2.304	1.651	1.918	2.317	0.779	0.725	0.300	2.618	2.183	1.305	2.131	1.221	1.227
7	2.675	1.907	2.278	2.732	0.833	0.841	0.353	3.227	2.547	1.609	2.694	1.407	1.417
8	2.528	1.807	2.111	2.565	0.809	0.784	0.324	2.881	2.373	1.467	2.451	1.352	1.358
9	2.374	1.686	2.020	2.406	0.769	0.740	0.303	2.705	2.235	1.389	2.280	1.249	1.255
10	2.748	1.946	2.264	2.782	0.849	0.834	0.342	3.209	2.549	1.589	2.654	1.435	1.446
11	2.422	1.737	2.002	2.522	0.757	0.737	0.301	2.728	2.274	1.374	2.298	1.262	1.271
12	2.224	1.603	1.812	2.274	0.729	0.653	0.261	2.480	2.083	1.233	2.084	1.176	1.179
13	2.389	1.693	2.053	2.476	0.744	0.767	0.314	2.750	2.239	1.416	2.344	1.268	1.273
14	2.467	1.763	2.054	2.423	0.750	0.740	0.308	2.872	2.383	1.464	2.350	1.295	1.296
15	2.270	1.610	1.908	2.320	0.774	0.720	0.289	2.595	2.150	1.327	2.183	1.219	1.219
16	2.440	1.709	2.045	2.496	0.849	0.746	0.302	2.768	2.254	1.410	2.334	1.310	1.307
17	2.461	1.732	2.070	2.456	0.735	0.759	0.321	2.837	2.268	1.420	2.324	1.270	1.284
18	2.385	1.712	1.976	2.389	0.728	0.731	0.303	2.777	2.220	1.374	2.369	1.248	1.256
19	2.438	1.747	2.052	2.442	0.731	0.734	0.309	2.824	2.323	1.429	2.450	1.272	1.276
20	2.181	1.568	1.803	2.217	0.670	0.676	0.278	2.556	2.071	1.211	2.063	1.160	1.169

Figura 45 – *Posthoc* de Shaffer para o Tempo no conjunto com $n = 60$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 54 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $n = 80$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.052	0.102	0.048	0.087	0.127	0.152	0.147	0.091	0.167	0.125	0.083	0.125	0.125
2	0.091	0.134	0.136	0.109	0.133	0.294	0.235	0.176	0.176	0.182	0.121	0.235	0.235
3	0.073	0.121	0.183	0.146	0.100	0.208	0.170	0.123	0.122	0.171	0.146	0.171	0.167
4	0.091	0.156	0.159	0.136	0.105	0.211	0.159	0.136	0.152	0.182	0.136	0.159	0.158
5	0.078	0.087	0.147	0.098	0.098	0.176	0.137	0.130	0.117	0.157	0.114	0.137	0.137
6	0.073	0.113	0.122	0.116	0.098	0.195	0.171	0.136	0.122	0.129	0.091	0.161	0.146
7	0.143	0.150	0.257	0.229	0.149	0.229	0.171	0.152	0.145	0.286	0.257	0.150	0.156
8	0.073	0.166	0.214	0.214	0.116	0.250	0.195	0.150	0.150	0.262	0.214	0.190	0.195
9	0.057	0.115	0.086	0.104	0.120	0.171	0.150	0.114	0.150	0.143	0.100	0.150	0.150
10	0.083	0.122	0.200	0.114	0.114	0.200	0.167	0.125	0.167	0.257	0.143	0.171	0.167
11	0.058	0.132	0.135	0.121	0.100	0.170	0.138	0.098	0.136	0.135	0.115	0.136	0.135
12	0.054	0.111	0.154	0.154	0.114	0.200	0.165	0.100	0.108	0.154	0.154	0.157	0.154
13	0.057	0.105	0.118	0.095	0.095	0.162	0.143	0.143	0.143	0.143	0.143	0.143	0.143
14	0.077	0.120	0.173	0.135	0.114	0.155	0.154	0.115	0.122	0.212	0.154	0.154	0.135
15	0.086	0.121	0.171	0.112	0.108	0.200	0.190	0.143	0.143	0.171	0.143	0.190	0.171
16	0.056	0.110	0.198	0.116	0.115	0.186	0.167	0.111	0.111	0.221	0.140	0.167	0.163
17	0.087	0.100	0.167	0.133	0.106	0.217	0.174	0.133	0.130	0.233	0.133	0.174	0.167
18	0.100	0.123	0.200	0.200	0.125	0.200	0.150	0.150	0.150	0.238	0.200	0.150	0.150
19	0.076	0.131	0.243	0.162	0.119	0.222	0.178	0.151	0.162	0.297	0.162	0.172	0.167
20	0.094	0.203	0.281	0.188	0.148	0.220	0.167	0.167	0.167	0.281	0.250	0.167	0.167

Figura 46 – *Posthoc* de Shaffer para $I_{\epsilon+}$ no conjunto com $n = 80$.

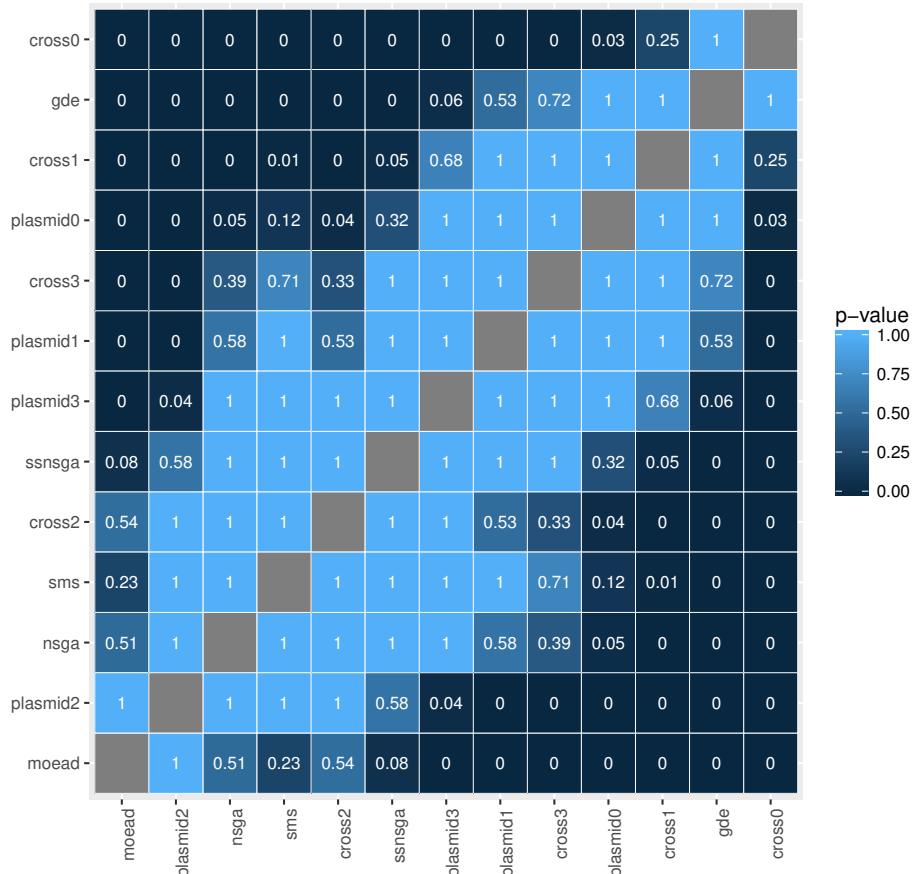
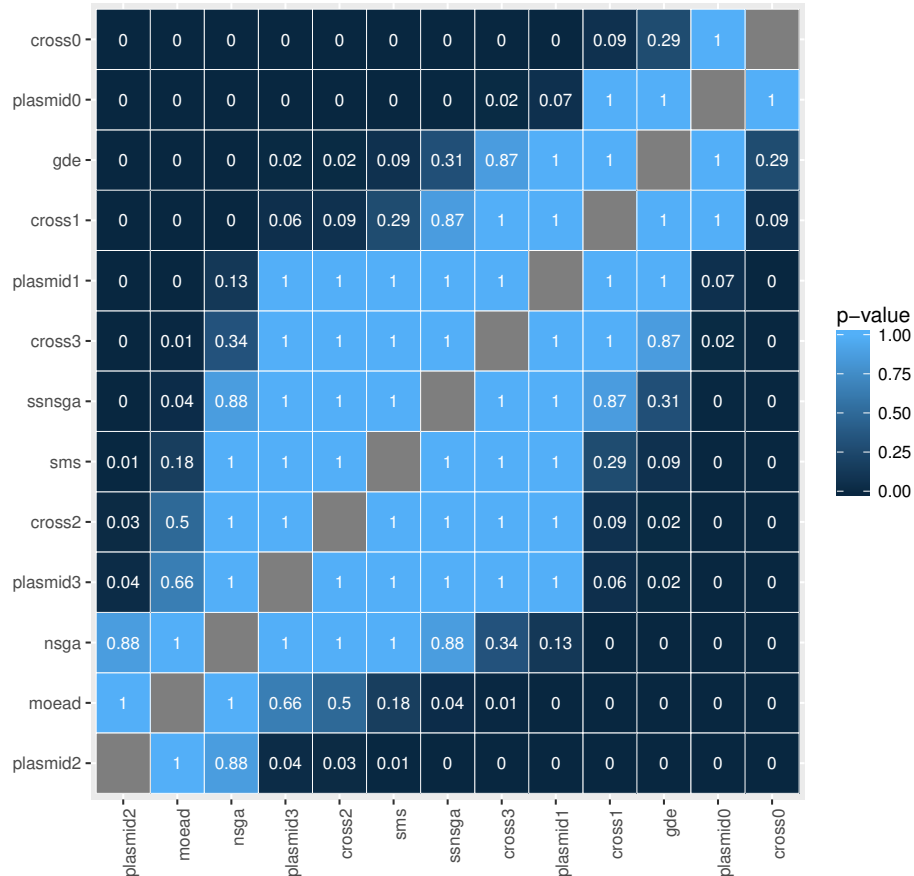


Tabela 55 – Medianas do indicador de qualidade I_H no conjunto com $n = 80$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.096	0.194	0.093	0.144	0.206	0.261	0.247	0.124	0.243	0.238	0.137	0.233	0.216
2	0.134	0.246	0.232	0.236	0.242	0.380	0.342	0.245	0.271	0.352	0.264	0.328	0.317
3	0.126	0.190	0.259	0.254	0.195	0.294	0.267	0.168	0.185	0.291	0.266	0.258	0.245
4	0.114	0.215	0.253	0.232	0.184	0.267	0.236	0.174	0.218	0.324	0.241	0.222	0.239
5	0.107	0.161	0.220	0.155	0.166	0.250	0.235	0.188	0.198	0.260	0.207	0.206	0.206
6	0.100	0.179	0.203	0.181	0.158	0.255	0.242	0.145	0.181	0.227	0.173	0.222	0.207
7	0.133	0.212	0.173	0.190	0.200	0.301	0.269	0.187	0.222	0.276	0.215	0.247	0.244
8	0.092	0.223	0.274	0.300	0.185	0.296	0.251	0.185	0.252	0.369	0.314	0.234	0.233
9	0.109	0.231	0.123	0.134	0.196	0.242	0.252	0.144	0.230	0.221	0.155	0.221	0.217
10	0.112	0.204	0.268	0.210	0.178	0.245	0.232	0.181	0.227	0.362	0.237	0.221	0.219
11	0.130	0.202	0.210	0.226	0.192	0.254	0.252	0.173	0.239	0.261	0.240	0.226	0.220
12	0.106	0.163	0.241	0.250	0.183	0.236	0.238	0.145	0.193	0.273	0.292	0.205	0.201
13	0.090	0.177	0.242	0.156	0.159	0.243	0.247	0.176	0.199	0.249	0.218	0.206	0.208
14	0.130	0.210	0.271	0.259	0.213	0.251	0.246	0.165	0.205	0.328	0.263	0.231	0.223
15	0.122	0.207	0.256	0.229	0.198	0.270	0.275	0.187	0.238	0.292	0.259	0.256	0.241
16	0.098	0.198	0.260	0.222	0.195	0.270	0.253	0.171	0.206	0.334	0.224	0.247	0.234
17	0.075	0.131	0.220	0.232	0.156	0.236	0.222	0.114	0.171	0.311	0.226	0.206	0.196
18	0.172	0.227	0.283	0.289	0.270	0.291	0.276	0.211	0.246	0.331	0.286	0.242	0.240
19	0.096	0.211	0.355	0.272	0.183	0.303	0.270	0.180	0.225	0.437	0.301	0.245	0.241
20	0.117	0.258	0.351	0.300	0.245	0.312	0.274	0.210	0.274	0.407	0.345	0.262	0.261

Figura 47 – *Posthoc* de Shaffer para I_H no conjunto com $n = 80$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 56 – Medianas do indicador de qualidade I_ϵ no conjunto com $n = 80$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.052	1.102	1.048	1.087	1.126	1.152	1.140	1.083	1.167	1.125	1.083	1.125	1.123
2	1.091	1.134	1.136	1.109	1.130	1.278	1.222	1.167	1.176	1.182	1.121	1.222	1.222
3	1.071	1.121	1.183	1.146	1.098	1.200	1.163	1.119	1.114	1.171	1.146	1.164	1.154
4	1.087	1.156	1.159	1.136	1.100	1.205	1.158	1.136	1.147	1.182	1.136	1.158	1.158
5	1.077	1.087	1.147	1.098	1.093	1.173	1.130	1.125	1.106	1.157	1.106	1.130	1.130
6	1.073	1.113	1.122	1.116	1.093	1.190	1.152	1.133	1.115	1.124	1.091	1.146	1.143
7	1.125	1.134	1.257	1.229	1.128	1.221	1.150	1.132	1.132	1.286	1.257	1.143	1.143
8	1.073	1.166	1.214	1.214	1.113	1.227	1.178	1.150	1.150	1.262	1.214	1.182	1.185
9	1.057	1.115	1.086	1.104	1.120	1.167	1.150	1.110	1.150	1.139	1.100	1.150	1.143
10	1.081	1.122	1.200	1.114	1.105	1.189	1.160	1.125	1.160	1.257	1.143	1.162	1.160
11	1.058	1.132	1.135	1.121	1.100	1.157	1.136	1.093	1.136	1.135	1.115	1.131	1.123
12	1.054	1.111	1.154	1.154	1.110	1.192	1.154	1.100	1.102	1.154	1.154	1.154	1.148
13	1.057	1.105	1.118	1.095	1.095	1.159	1.143	1.143	1.143	1.143	1.143	1.143	1.136
14	1.077	1.120	1.173	1.135	1.114	1.150	1.139	1.115	1.115	1.212	1.154	1.134	1.125
15	1.086	1.121	1.171	1.112	1.108	1.200	1.182	1.136	1.143	1.171	1.136	1.182	1.171
16	1.056	1.110	1.198	1.116	1.111	1.176	1.167	1.111	1.111	1.221	1.140	1.167	1.154
17	1.087	1.100	1.167	1.133	1.105	1.217	1.174	1.130	1.125	1.233	1.133	1.174	1.167
18	1.100	1.123	1.200	1.200	1.119	1.190	1.150	1.142	1.150	1.238	1.200	1.133	1.133
19	1.071	1.131	1.243	1.162	1.119	1.215	1.167	1.139	1.146	1.297	1.162	1.167	1.167
20	1.087	1.203	1.281	1.188	1.148	1.200	1.167	1.158	1.167	1.281	1.250	1.167	1.167

Figura 48 – *Posthoc* de Shaffer para I_ϵ no conjunto com $n = 80$.

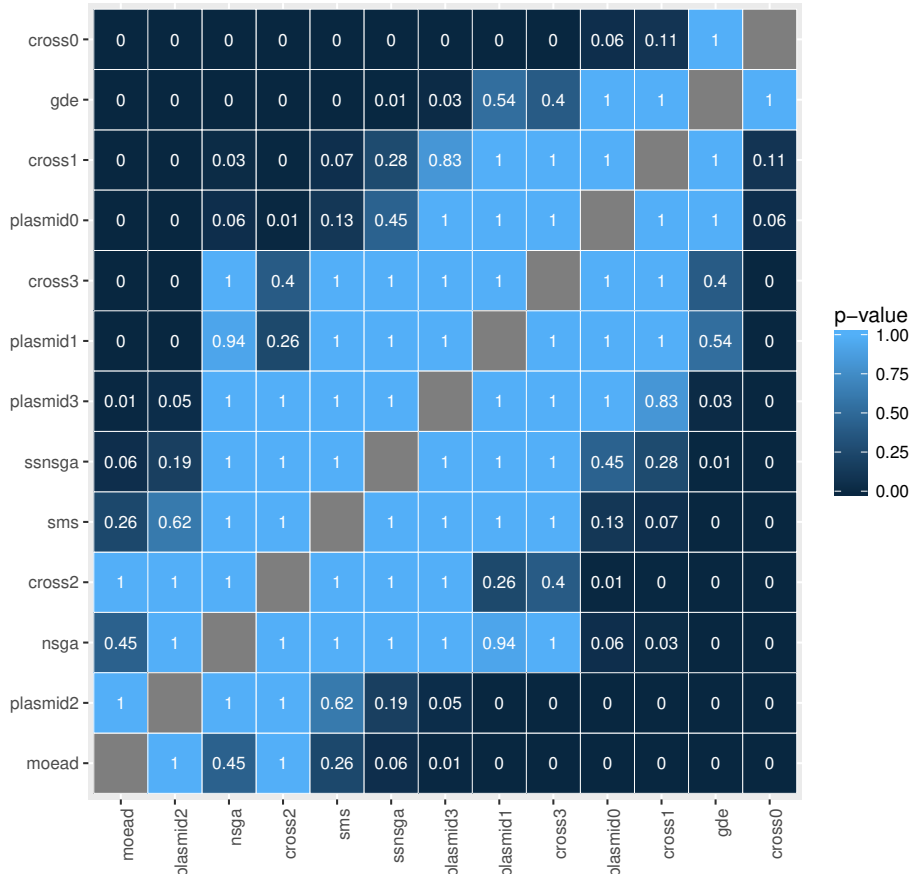
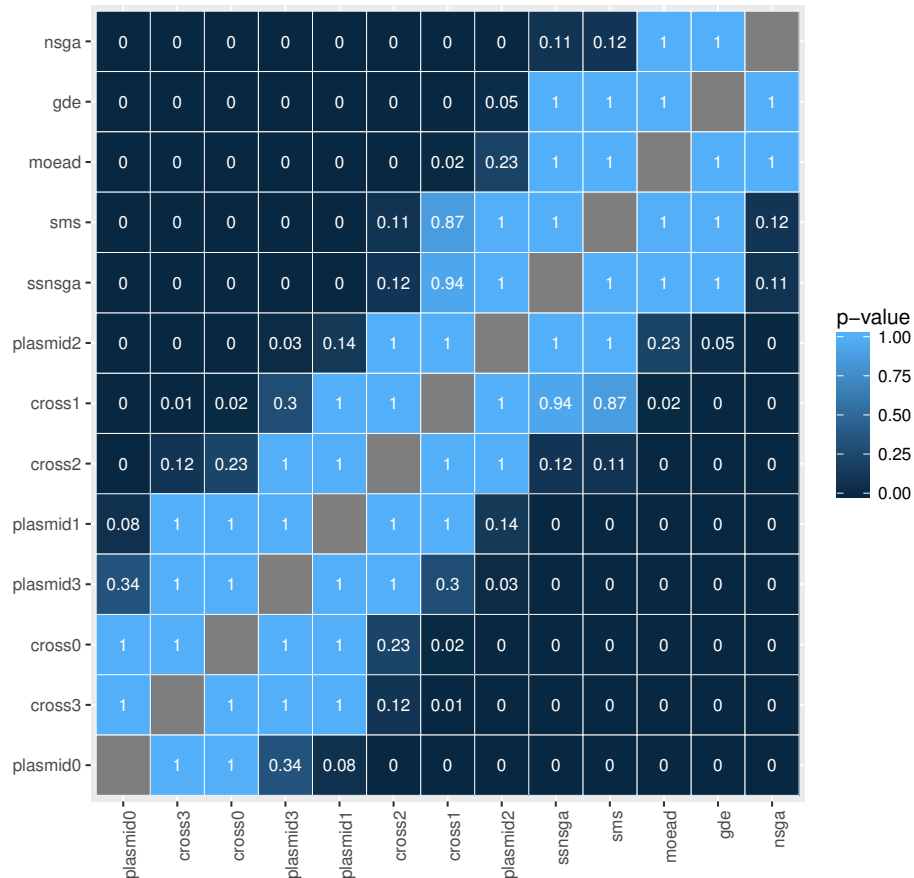


Tabela 57 – Medianas do tempo de processamento no conjunto com $n = 80$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	2.972	2.108	2.510	2.974	1.018	1.021	0.430	3.357	2.697	1.758	2.800	1.535	1.538
2	2.544	1.821	2.137	2.557	0.778	0.898	0.363	2.936	2.442	1.459	2.428	1.335	1.340
3	2.880	2.033	2.418	2.882	0.860	1.012	0.408	3.382	2.740	1.713	2.881	1.476	1.481
4	2.768	1.927	2.321	2.783	0.795	0.994	0.406	3.244	2.588	1.644	2.757	1.408	1.419
5	2.518	1.818	2.087	2.606	0.736	0.878	0.346	2.857	2.427	1.453	2.449	1.312	1.310
6	2.566	1.795	2.169	2.552	0.767	0.896	0.354	2.958	2.403	1.464	2.423	1.318	1.315
7	2.606	1.858	2.192	2.590	0.775	0.899	0.375	2.997	2.526	1.489	2.474	1.323	1.335
8	2.799	1.993	2.343	2.759	0.823	0.951	0.396	3.286	2.633	1.651	2.815	1.445	1.455
9	2.532	1.772	2.140	2.630	0.818	0.864	0.337	2.923	2.319	1.465	2.491	1.317	1.321
10	2.525	1.798	2.106	2.561	0.754	0.846	0.340	2.888	2.411	1.437	2.469	1.293	1.297
11	2.348	1.695	1.970	2.398	0.719	0.828	0.319	2.692	2.212	1.367	2.272	1.243	1.244
12	2.378	1.718	2.038	2.462	0.732	0.834	0.330	2.759	2.295	1.391	2.279	1.261	1.261
13	2.518	1.793	2.143	2.569	0.765	0.862	0.340	2.887	2.337	1.480	2.411	1.305	1.310
14	2.639	1.926	2.142	2.610	0.791	0.908	0.376	3.094	2.545	1.501	2.513	1.385	1.398
15	2.652	1.901	2.236	2.692	0.837	0.893	0.359	3.070	2.503	1.565	2.577	1.400	1.402
16	2.540	1.824	2.086	2.561	0.766	0.877	0.351	2.887	2.395	1.432	2.418	1.338	1.340
17	2.280	1.641	1.941	2.302	0.730	0.783	0.307	2.649	2.159	1.325	2.255	1.222	1.221
18	2.468	1.801	2.035	2.462	0.756	0.841	0.339	2.754	2.340	1.368	2.299	1.307	1.309
19	2.479	1.800	2.062	2.473	0.761	0.838	0.335	2.849	2.417	1.411	2.365	1.298	1.302
20	2.994	2.132	2.542	2.937	0.883	1.018	0.429	3.471	2.828	1.806	2.895	1.528	1.542

Figura 49 – *Posthoc* de Shaffer para o Tempo no conjunto com $n = 80$.

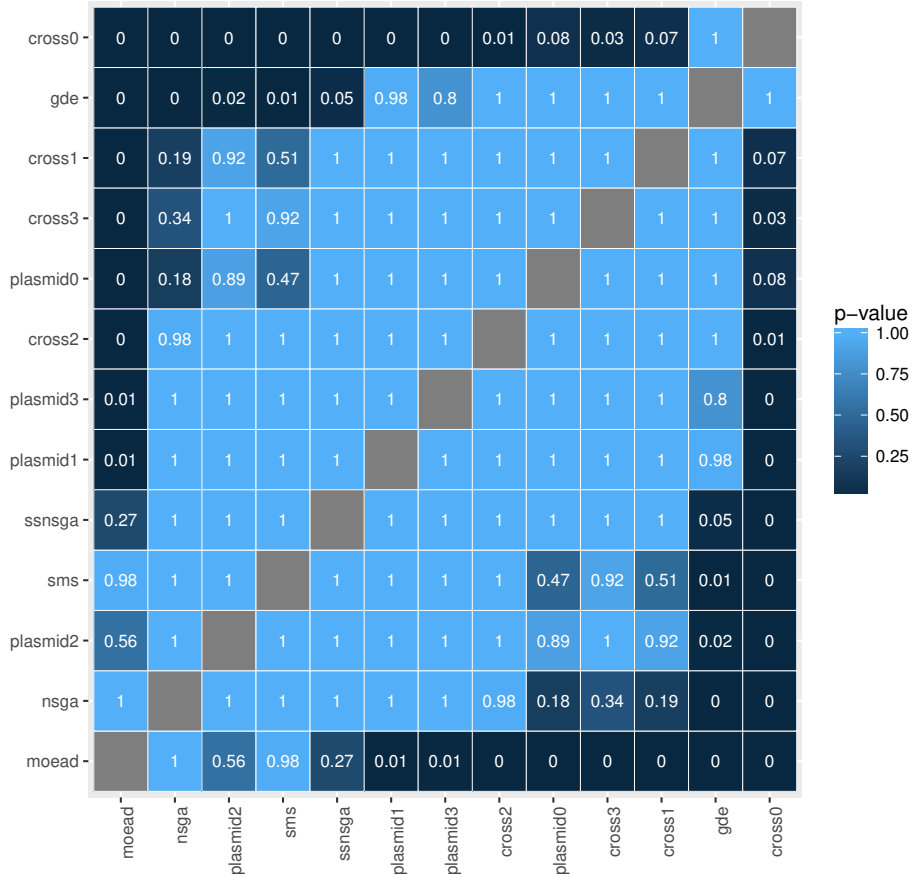


C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 58 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $n = 100$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.083	0.130	0.083	0.114	0.124	0.172	0.152	0.125	0.125	0.136	0.125	0.147	0.147
2	0.081	0.118	0.162	0.162	0.095	0.192	0.153	0.122	0.108	0.189	0.162	0.162	0.138
3	0.083	0.096	0.143	0.143	0.116	0.200	0.167	0.125	0.125	0.155	0.143	0.167	0.163
4	0.071	0.114	0.071	0.101	0.111	0.190	0.143	0.120	0.146	0.117	0.100	0.143	0.143
5	0.100	0.132	0.257	0.214	0.114	0.250	0.200	0.161	0.150	0.257	0.229	0.200	0.194
6	0.100	0.133	0.263	0.132	0.109	0.258	0.200	0.171	0.171	0.263	0.184	0.200	0.193
7	0.065	0.117	0.217	0.152	0.109	0.182	0.158	0.136	0.136	0.239	0.174	0.136	0.136
8	0.081	0.141	0.230	0.216	0.133	0.189	0.158	0.135	0.135	0.243	0.216	0.135	0.135
9	0.080	0.115	0.080	0.094	0.094	0.200	0.188	0.120	0.156	0.120	0.120	0.160	0.156
10	0.080	0.107	0.243	0.243	0.108	0.189	0.160	0.120	0.120	0.270	0.243	0.160	0.135
11	0.122	0.146	0.073	0.110	0.172	0.293	0.227	0.146	0.158	0.091	0.091	0.227	0.220
12	0.068	0.115	0.205	0.136	0.100	0.200	0.159	0.114	0.150	0.205	0.150	0.150	0.150
13	0.057	0.142	0.043	0.104	0.108	0.194	0.161	0.129	0.130	0.130	0.113	0.158	0.161
14	0.077	0.126	0.095	0.127	0.117	0.214	0.190	0.143	0.115	0.143	0.095	0.179	0.167
15	0.077	0.097	0.103	0.103	0.110	0.205	0.154	0.106	0.100	0.115	0.100	0.154	0.145
16	0.069	0.125	0.154	0.179	0.113	0.192	0.158	0.128	0.128	0.158	0.154	0.154	0.154
17	0.233	0.751	0.107	0.098	0.242	0.270	0.230	0.240	0.750	0.130	0.087	0.236	0.230
18	0.257	0.796	0.059	0.113	0.243	0.294	0.243	0.244	0.799	0.130	0.118	0.241	0.241
19	0.067	0.126	0.111	0.092	0.113	0.206	0.200	0.150	0.150	0.150	0.133	0.178	0.161
20	0.061	0.125	0.136	0.119	0.131	0.227	0.186	0.136	0.136	0.136	0.205	0.182	0.182

Figura 50 – *Posthoc* de Shaffer para I_{ϵ_+} no conjunto com $n = 100$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 60 – Medianas do indicador de qualidade I_ϵ no conjunto com $n = 100$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.083	1.130	1.083	1.114	1.123	1.169	1.147	1.125	1.125	1.134	1.119	1.147	1.147
2	1.081	1.118	1.162	1.162	1.091	1.189	1.140	1.117	1.108	1.189	1.162	1.162	1.135
3	1.077	1.096	1.143	1.143	1.116	1.191	1.160	1.125	1.125	1.151	1.143	1.162	1.149
4	1.071	1.114	1.071	1.101	1.111	1.178	1.139	1.118	1.142	1.107	1.100	1.133	1.133
5	1.087	1.132	1.257	1.214	1.114	1.228	1.171	1.150	1.150	1.257	1.229	1.182	1.171
6	1.091	1.132	1.263	1.132	1.105	1.244	1.182	1.169	1.158	1.263	1.184	1.182	1.179
7	1.065	1.117	1.217	1.152	1.102	1.168	1.140	1.136	1.136	1.239	1.174	1.136	1.136
8	1.077	1.141	1.230	1.216	1.133	1.175	1.133	1.125	1.125	1.243	1.216	1.125	1.125
9	1.077	1.115	1.080	1.094	1.094	1.192	1.186	1.120	1.147	1.120	1.120	1.156	1.156
10	1.074	1.107	1.243	1.243	1.105	1.171	1.143	1.115	1.120	1.270	1.243	1.143	1.123
11	1.122	1.135	1.073	1.110	1.163	1.273	1.226	1.141	1.151	1.091	1.091	1.217	1.210
12	1.068	1.115	1.205	1.136	1.100	1.174	1.150	1.104	1.150	1.205	1.150	1.146	1.147
13	1.054	1.142	1.043	1.104	1.104	1.188	1.156	1.125	1.130	1.130	1.107	1.148	1.152
14	1.077	1.126	1.095	1.127	1.115	1.214	1.181	1.138	1.115	1.143	1.095	1.173	1.167
15	1.071	1.097	1.103	1.103	1.106	1.205	1.150	1.103	1.100	1.111	1.095	1.146	1.137
16	1.068	1.125	1.154	1.179	1.113	1.185	1.150	1.123	1.122	1.158	1.154	1.150	1.150
17	1.219	1.751	1.107	1.098	1.221	1.268	1.215	1.222	1.750	1.130	1.087	1.219	1.216
18	1.235	1.796	1.059	1.113	1.235	1.277	1.235	1.235	1.799	1.118	1.118	1.232	1.234
19	1.067	1.126	1.111	1.092	1.113	1.200	1.190	1.143	1.146	1.150	1.132	1.178	1.156
20	1.060	1.124	1.136	1.119	1.123	1.217	1.182	1.136	1.136	1.136	1.201	1.182	1.176

Figura 52 – *Posthoc* de Shaffer para I_ϵ no conjunto com $n = 100$.

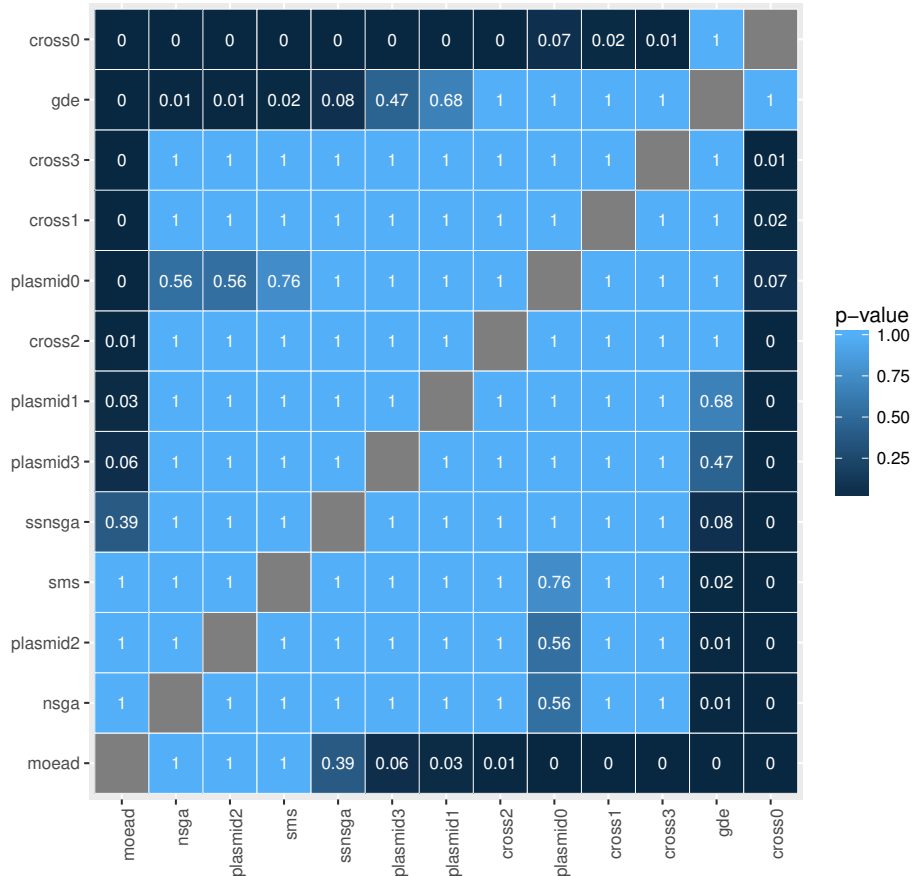
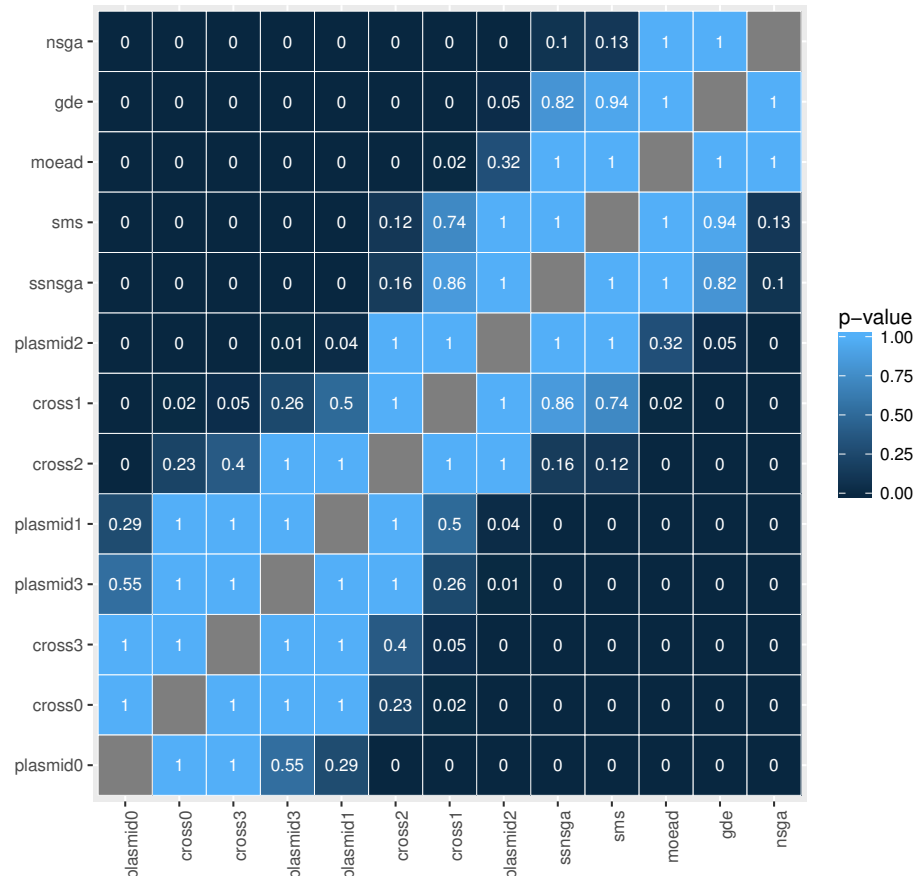


Tabela 61 – Medianas do tempo de processamento no conjunto com $n = 100$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	2.750	1.971	2.369	2.794	0.802	1.018	0.400	3.370	2.628	1.645	2.871	1.414	1.420
2	2.402	1.728	2.042	2.389	0.707	0.906	0.338	2.734	2.301	1.400	2.286	1.256	1.262
3	2.668	1.944	2.226	2.684	0.787	0.990	0.389	3.129	2.565	1.569	2.527	1.393	1.401
4	2.699	1.917	2.293	2.710	0.809	1.004	0.399	3.084	2.572	1.585	2.650	1.378	1.384
5	2.673	1.913	2.222	2.604	0.801	0.982	0.387	3.218	2.536	1.540	2.682	1.348	1.356
6	2.803	2.013	2.326	2.707	0.790	1.045	0.418	3.221	2.649	1.606	2.579	1.405	1.413
7	2.688	1.934	2.226	2.687	0.772	0.970	0.380	3.151	2.571	1.545	2.667	1.371	1.382
8	2.897	2.111	2.398	2.858	0.828	1.044	0.430	3.371	2.798	1.613	3.062	1.479	1.494
9	2.848	2.067	2.398	2.822	0.836	1.047	0.409	3.338	2.747	1.662	2.689	1.489	1.490
10	2.560	1.863	2.079	2.540	0.738	0.913	0.356	2.929	2.414	1.407	2.408	1.329	1.334
11	2.818	1.993	2.374	2.792	0.800	1.063	0.419	3.301	2.702	1.637	2.808	1.442	1.443
12	2.277	1.662	1.896	2.289	0.716	0.868	0.315	2.607	2.174	1.308	2.170	1.212	1.218
13	2.704	1.934	2.280	2.788	0.801	1.003	0.393	3.197	2.596	1.598	2.763	1.396	1.397
14	2.767	1.948	2.346	2.724	0.845	1.032	0.404	3.280	2.615	1.665	2.769	1.451	1.453
15	1.966	1.439	1.637	1.953	0.611	0.765	0.270	2.217	1.887	1.131	1.808	1.050	1.052
16	2.897	2.104	2.429	2.898	0.863	1.116	0.446	3.329	2.817	1.681	2.779	1.510	1.518
17	2.989	2.463	2.279	2.706	0.801	1.000	0.427	3.408	3.186	1.568	2.578	1.422	1.441
18	3.398	2.654	2.606	3.132	0.833	1.184	0.537	3.887	3.495	1.889	3.125	1.565	1.582
19	2.647	1.869	2.267	2.657	0.785	0.990	0.381	3.035	2.480	1.528	2.554	1.346	1.356
20	2.715	1.964	2.299	2.683	0.842	1.005	0.401	3.153	2.631	1.567	2.468	1.414	1.415

Figura 53 – *Posthoc* de Shaffer para o Tempo no conjunto com $n = 100$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 62 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $D^k = 10$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.087	0.159	0.087	0.115	0.116	0.227	0.182	0.144	0.169	0.143	0.091	0.182	0.182
2	0.087	0.163	0.238	0.250	0.134	0.217	0.174	0.130	0.130	0.250	0.225	0.174	0.150
3	0.069	0.125	0.172	0.121	0.109	0.182	0.144	0.136	0.136	0.207	0.164	0.138	0.138
4	0.133	0.133	0.133	0.111	0.133	0.167	0.167	0.133	0.133	0.167	0.125	0.156	0.156
5	0.077	0.136	0.200	0.200	0.111	0.178	0.156	0.130	0.130	0.200	0.200	0.133	0.133
6	0.237	0.771	0.125	0.109	0.234	0.278	0.234	0.234	0.773	0.125	0.125	0.237	0.234
7	0.064	0.152	0.185	0.204	0.104	0.204	0.136	0.136	0.111	0.204	0.185	0.136	0.136
8	0.075	0.132	0.245	0.176	0.124	0.176	0.157	0.133	0.130	0.255	0.216	0.157	0.137
9	0.091	0.137	0.094	0.104	0.106	0.189	0.170	0.136	0.136	0.136	0.094	0.151	0.144
10	0.081	0.117	0.216	0.189	0.115	0.189	0.162	0.150	0.135	0.216	0.189	0.162	0.156
11	0.069	0.105	0.147	0.147	0.102	0.176	0.154	0.125	0.125	0.147	0.147	0.149	0.147
12	0.070	0.170	0.163	0.118	0.112	0.163	0.140	0.116	0.118	0.209	0.140	0.139	0.136
13	0.076	0.157	0.250	0.225	0.116	0.208	0.150	0.125	0.125	0.275	0.225	0.167	0.150
14	0.089	0.117	0.178	0.133	0.091	0.182	0.156	0.136	0.136	0.178	0.156	0.156	0.136
15	0.105	0.125	0.104	0.111	0.146	0.263	0.227	0.158	0.167	0.208	0.105	0.226	0.211
16	0.080	0.154	0.207	0.138	0.127	0.160	0.138	0.120	0.160	0.207	0.172	0.121	0.121
17	0.081	0.153	0.324	0.243	0.108	0.189	0.136	0.136	0.136	0.297	0.270	0.136	0.136
18	0.069	0.130	0.148	0.148	0.103	0.209	0.148	0.114	0.111	0.148	0.148	0.167	0.148
19	0.080	0.151	0.093	0.115	0.110	0.200	0.160	0.120	0.147	0.160	0.120	0.160	0.149
20	0.105	0.157	0.156	0.131	0.111	0.222	0.178	0.158	0.163	0.158	0.156	0.178	0.158

Figura 54 – *Posthoc* de Shaffer para I_{ϵ_+} no conjunto com $D^k = 10$.

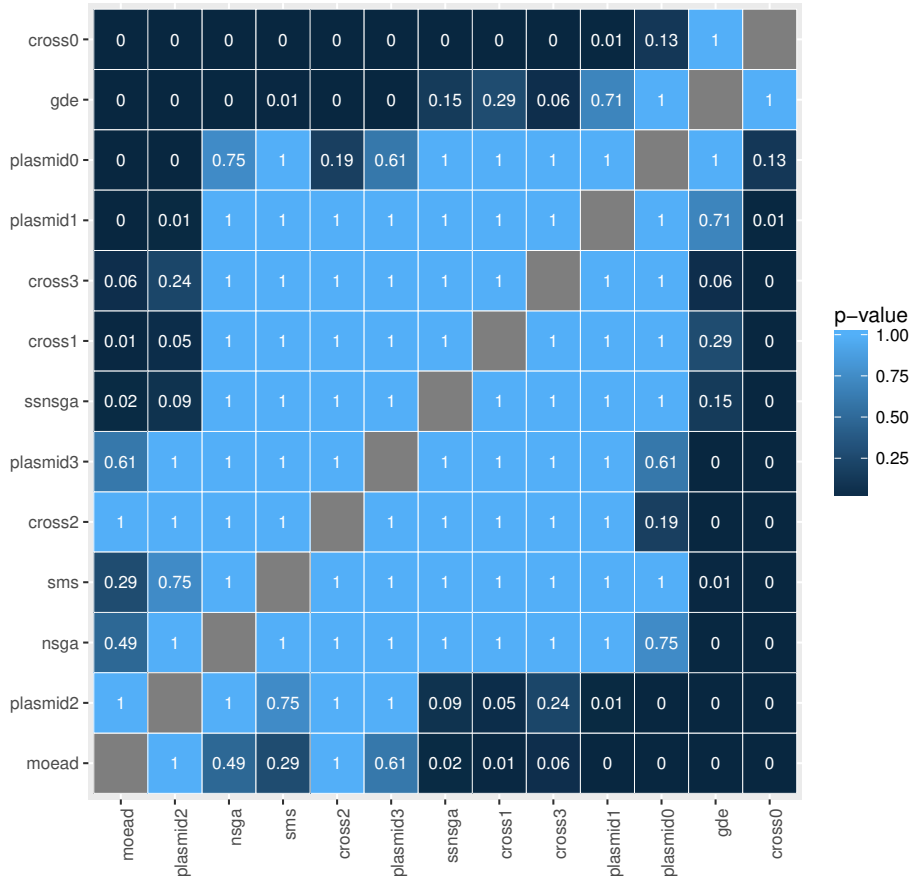
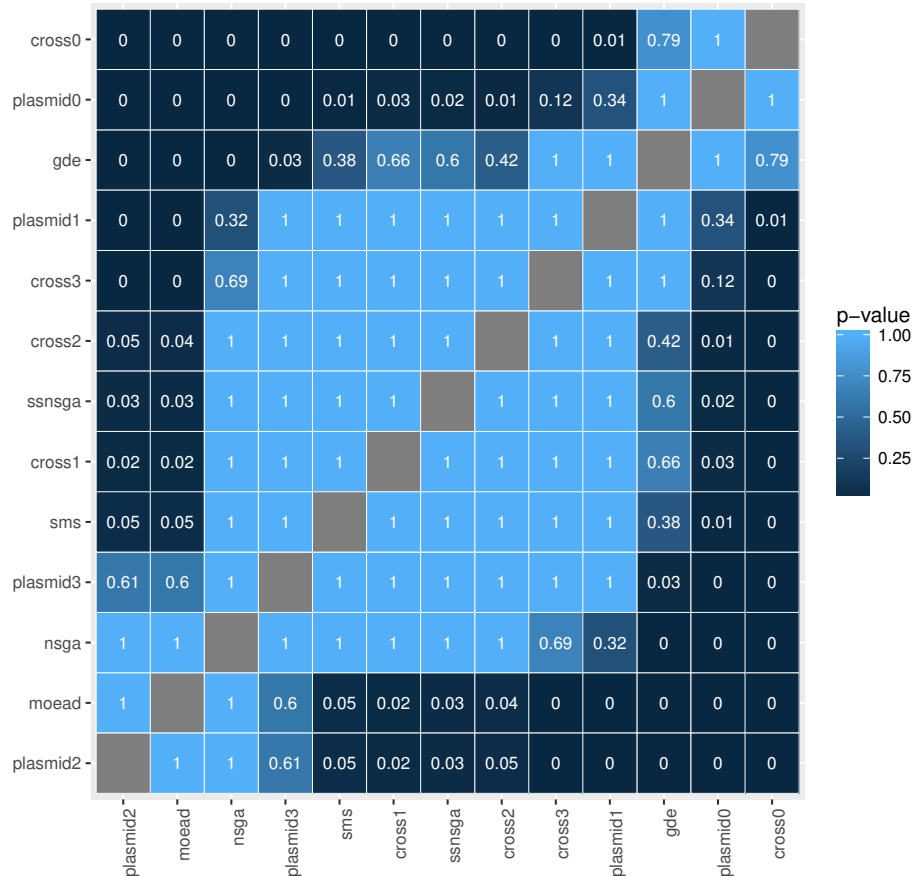


Tabela 63 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 10$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.113	0.274	0.172	0.145	0.196	0.303	0.260	0.180	0.244	0.241	0.127	0.254	0.249
2	0.131	0.244	0.289	0.317	0.220	0.285	0.261	0.172	0.197	0.325	0.306	0.236	0.238
3	0.093	0.190	0.203	0.191	0.180	0.240	0.222	0.169	0.200	0.292	0.270	0.213	0.214
4	0.234	0.313	0.230	0.181	0.281	0.313	0.319	0.267	0.315	0.250	0.216	0.280	0.281
5	0.109	0.226	0.279	0.281	0.212	0.265	0.256	0.182	0.197	0.331	0.307	0.220	0.219
6	0.416	1.011	0.130	0.134	0.331	0.491	0.410	0.369	1.026	0.170	0.172	0.411	0.412
7	0.116	0.235	0.258	0.316	0.191	0.270	0.258	0.185	0.182	0.328	0.293	0.228	0.227
8	0.120	0.212	0.259	0.237	0.222	0.270	0.249	0.182	0.189	0.328	0.269	0.231	0.226
9	0.108	0.205	0.183	0.150	0.178	0.278	0.229	0.153	0.187	0.256	0.176	0.222	0.216
10	0.142	0.226	0.280	0.239	0.224	0.289	0.285	0.196	0.212	0.315	0.285	0.258	0.253
11	0.109	0.185	0.228	0.258	0.180	0.257	0.247	0.168	0.207	0.298	0.264	0.233	0.222
12	0.101	0.231	0.209	0.200	0.162	0.247	0.247	0.170	0.234	0.302	0.245	0.218	0.209
13	0.132	0.209	0.296	0.283	0.218	0.286	0.254	0.191	0.197	0.369	0.297	0.231	0.226
14	0.106	0.181	0.253	0.226	0.155	0.261	0.225	0.146	0.180	0.306	0.239	0.222	0.214
15	0.224	0.306	0.186	0.192	0.296	0.395	0.373	0.278	0.354	0.383	0.226	0.333	0.353
16	0.160	0.266	0.268	0.230	0.246	0.286	0.258	0.190	0.235	0.300	0.251	0.243	0.239
17	0.117	0.219	0.319	0.262	0.200	0.267	0.238	0.182	0.207	0.329	0.283	0.213	0.219
18	0.107	0.207	0.207	0.221	0.161	0.278	0.231	0.152	0.169	0.240	0.249	0.216	0.212
19	0.149	0.243	0.193	0.227	0.215	0.276	0.249	0.190	0.223	0.255	0.223	0.239	0.239
20	0.155	0.273	0.262	0.221	0.221	0.340	0.312	0.229	0.274	0.302	0.270	0.289	0.292

Figura 55 – *Posthoc* de Shaffer para I_H no conjunto com $D^k = 10$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 64 – Medianas do indicador de qualidade I_ϵ no conjunto com $D^k = 10$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.083	1.159	1.087	1.115	1.113	1.227	1.178	1.138	1.165	1.135	1.089	1.178	1.178
2	1.078	1.163	1.238	1.250	1.134	1.192	1.154	1.122	1.115	1.250	1.225	1.154	1.136
3	1.068	1.125	1.172	1.121	1.107	1.171	1.136	1.135	1.136	1.207	1.164	1.136	1.136
4	1.133	1.133	1.133	1.111	1.133	1.163	1.154	1.133	1.133	1.160	1.118	1.155	1.149
5	1.068	1.136	1.200	1.200	1.103	1.167	1.143	1.125	1.130	1.200	1.200	1.128	1.125
6	1.224	1.768	1.125	1.109	1.214	1.257	1.214	1.218	1.771	1.125	1.125	1.225	1.214
7	1.064	1.152	1.185	1.204	1.104	1.200	1.136	1.133	1.101	1.204	1.185	1.136	1.136
8	1.066	1.132	1.245	1.176	1.120	1.174	1.143	1.130	1.130	1.255	1.216	1.136	1.130
9	1.087	1.137	1.094	1.104	1.106	1.185	1.164	1.136	1.136	1.136	1.091	1.145	1.136
10	1.081	1.117	1.216	1.189	1.115	1.184	1.154	1.128	1.128	1.216	1.189	1.154	1.148
11	1.069	1.105	1.147	1.147	1.100	1.173	1.147	1.125	1.125	1.147	1.147	1.147	1.138
12	1.070	1.170	1.163	1.118	1.112	1.156	1.136	1.111	1.114	1.209	1.140	1.136	1.136
13	1.074	1.157	1.250	1.225	1.111	1.186	1.136	1.125	1.125	1.275	1.225	1.143	1.133
14	1.083	1.117	1.178	1.133	1.089	1.178	1.156	1.128	1.136	1.178	1.156	1.156	1.136
15	1.105	1.125	1.104	1.111	1.144	1.250	1.211	1.158	1.161	1.208	1.105	1.211	1.204
16	1.080	1.154	1.207	1.138	1.127	1.149	1.125	1.120	1.160	1.207	1.172	1.120	1.120
17	1.081	1.153	1.324	1.243	1.097	1.167	1.136	1.132	1.136	1.297	1.270	1.121	1.133
18	1.066	1.130	1.148	1.148	1.103	1.200	1.143	1.109	1.103	1.148	1.148	1.154	1.136
19	1.080	1.150	1.093	1.115	1.107	1.185	1.151	1.120	1.134	1.160	1.120	1.145	1.143
20	1.098	1.157	1.156	1.131	1.108	1.214	1.162	1.158	1.158	1.158	1.156	1.163	1.158

Figura 56 – *Posthoc* de Shaffer para I_ϵ no conjunto com $D^k = 10$.

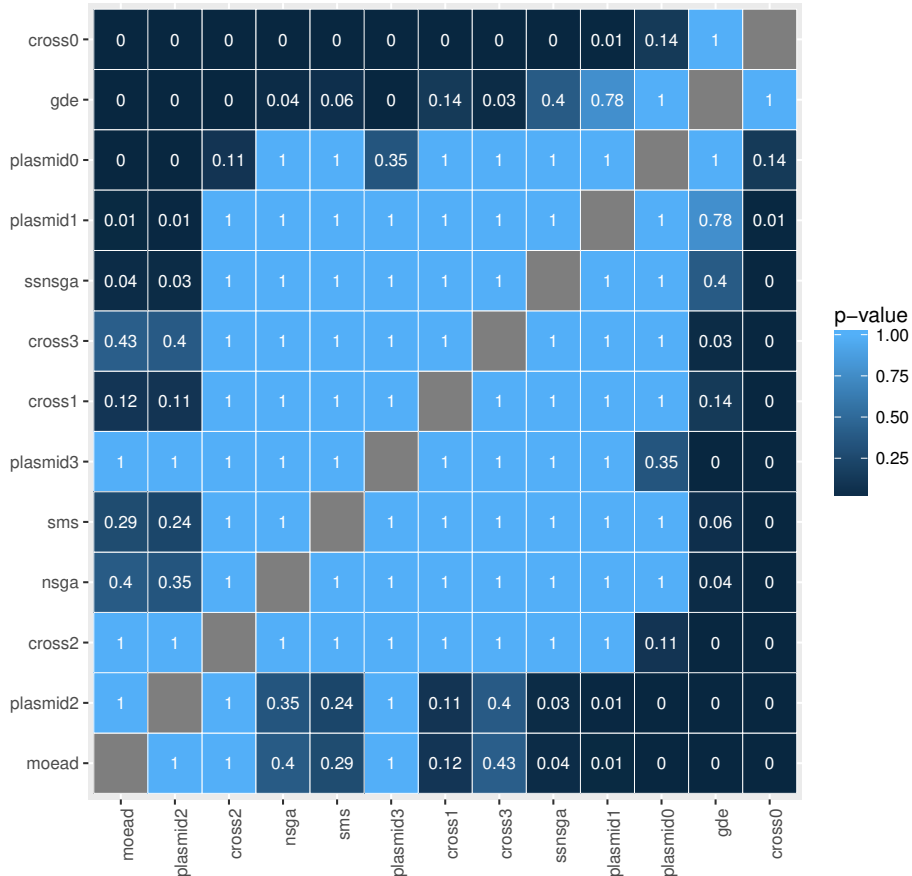
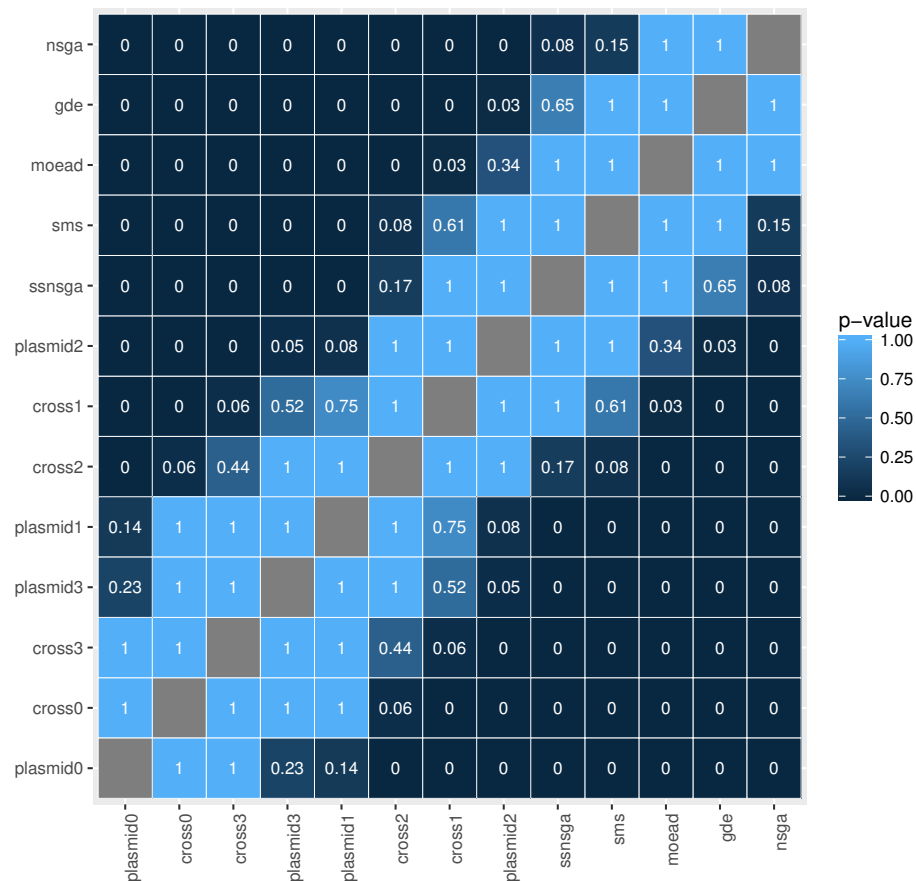


Tabela 65 – Medianas do tempo de processamento no conjunto com $D^k = 10$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	2.784	1.974	2.316	2.750	0.822	1.022	0.403	3.329	2.667	1.610	2.803	1.433	1.440
2	2.784	2.036	2.235	2.723	0.809	1.029	0.407	3.212	2.701	1.534	2.559	1.426	1.445
3	2.786	2.034	2.277	2.797	0.847	1.029	0.403	3.243	2.672	1.609	2.688	1.430	1.451
4	2.816	1.986	2.378	2.814	0.869	1.085	0.423	3.285	2.644	1.674	2.840	1.431	1.445
5	2.803	2.055	2.300	2.754	0.837	1.042	0.421	3.230	2.727	1.589	2.631	1.449	1.466
6	3.008	2.434	2.304	2.749	0.813	1.020	0.436	3.319	3.057	1.578	2.539	1.448	1.452
7	2.790	2.038	2.289	2.752	0.867	1.039	0.409	3.146	2.671	1.576	2.542	1.437	1.446
8	2.834	2.063	2.317	2.756	0.900	1.044	0.415	3.321	2.736	1.593	2.763	1.440	1.454
9	2.755	1.952	2.334	2.791	0.886	0.989	0.388	3.289	2.616	1.608	2.780	1.410	1.426
10	2.686	1.931	2.233	2.698	0.888	1.008	0.382	3.097	2.585	1.528	2.592	1.412	1.415
11	2.702	1.941	2.265	2.666	0.882	1.005	0.388	3.102	2.566	1.552	2.548	1.405	1.413
12	2.724	1.933	2.294	2.712	0.873	1.033	0.395	3.172	2.568	1.604	2.709	1.406	1.421
13	2.876	2.075	2.358	2.805	0.910	1.061	0.417	3.369	2.777	1.645	2.750	1.469	1.483
14	2.925	2.115	2.478	2.835	0.917	1.103	0.433	3.583	2.823	1.690	2.845	1.454	1.469
15	2.808	1.971	2.358	2.761	0.914	1.080	0.420	3.257	2.661	1.640	2.770	1.438	1.444
16	2.832	2.056	2.335	2.800	0.928	1.031	0.413	3.257	2.646	1.584	2.725	1.449	1.468
17	2.810	2.029	2.328	2.720	0.934	1.009	0.391	3.358	2.680	1.607	2.713	1.430	1.442
18	2.863	2.073	2.364	2.800	0.920	1.081	0.430	3.289	2.763	1.636	2.682	1.455	1.474
19	2.759	1.953	2.323	2.782	0.928	0.982	0.375	3.196	2.571	1.562	2.643	1.405	1.415
20	2.806	1.998	2.380	2.735	0.905	1.064	0.410	3.280	2.702	1.636	2.551	1.433	1.446

Figura 57 – *Posthoc* de Shaffer para o Tempo no conjunto com $D^k = 10$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 66 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $D^k = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.091	0.206	0.200	0.164	0.155	0.238	0.190	0.190	0.210	0.236	0.182	0.190	0.190
2	0.095	0.237	0.286	0.196	0.166	0.196	0.190	0.190	0.238	0.286	0.232	0.170	0.185
3	0.100	0.215	0.200	0.161	0.200	0.156	0.148	0.148	0.204	0.222	0.178	0.148	0.148
4	0.098	0.173	0.195	0.132	0.156	0.244	0.182	0.179	0.214	0.244	0.146	0.179	0.179
5	0.077	0.157	0.330	0.227	0.117	0.187	0.154	0.154	0.154	0.352	0.227	0.154	0.145
6	0.132	0.158	0.171	0.168	0.179	0.237	0.211	0.185	0.222	0.237	0.158	0.198	0.185
7	0.196	0.196	0.118	0.167	0.235	0.294	0.275	0.235	0.286	0.206	0.190	0.275	0.275
8	0.095	0.179	0.158	0.167	0.177	0.238	0.237	0.190	0.238	0.263	0.211	0.190	0.190
9	0.083	0.198	0.271	0.229	0.161	0.208	0.185	0.185	0.222	0.333	0.271	0.185	0.185
10	0.095	0.215	0.240	0.180	0.203	0.217	0.180	0.174	0.217	0.280	0.220	0.180	0.177
11	0.107	0.217	0.220	0.195	0.201	0.244	0.217	0.217	0.261	0.268	0.220	0.217	0.217
12	0.094	0.225	0.222	0.200	0.173	0.227	0.191	0.182	0.227	0.311	0.222	0.200	0.182
13	0.109	0.136	0.144	0.122	0.166	0.304	0.270	0.227	0.227	0.217	0.182	0.261	0.239
14	0.135	0.206	0.135	0.171	0.164	0.273	0.243	0.216	0.250	0.216	0.182	0.243	0.227
15	0.092	0.213	0.180	0.190	0.173	0.180	0.179	0.160	0.179	0.240	0.200	0.143	0.169
16	0.091	0.163	0.125	0.159	0.167	0.219	0.219	0.182	0.227	0.188	0.156	0.188	0.188
17	0.097	0.200	0.158	0.158	0.143	0.294	0.263	0.211	0.316	0.235	0.176	0.235	0.263
18	0.095	0.233	0.167	0.179	0.168	0.208	0.167	0.167	0.250	0.185	0.167	0.167	0.167
19	0.097	0.207	0.217	0.181	0.152	0.217	0.174	0.174	0.185	0.304	0.174	0.196	0.174
20	0.096	0.222	0.231	0.173	0.190	0.238	0.212	0.215	0.286	0.269	0.231	0.192	0.192

Figura 58 – *Posthoc* de Shaffer para I_{ϵ_+} no conjunto com $D^k = 20$.

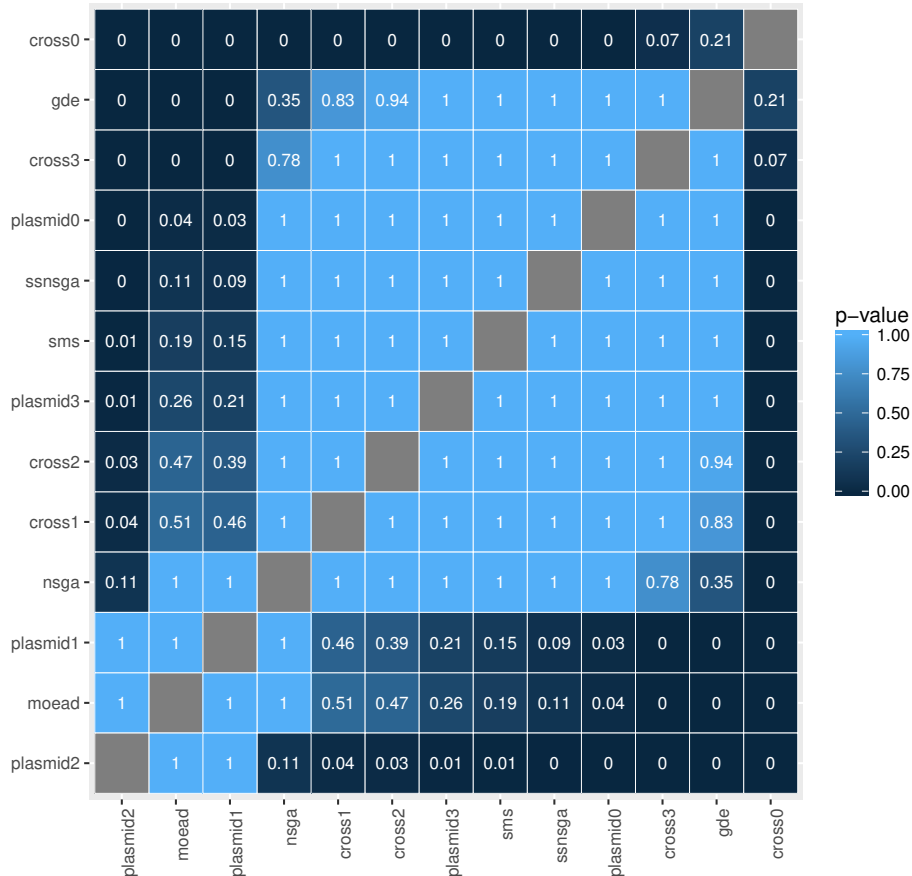
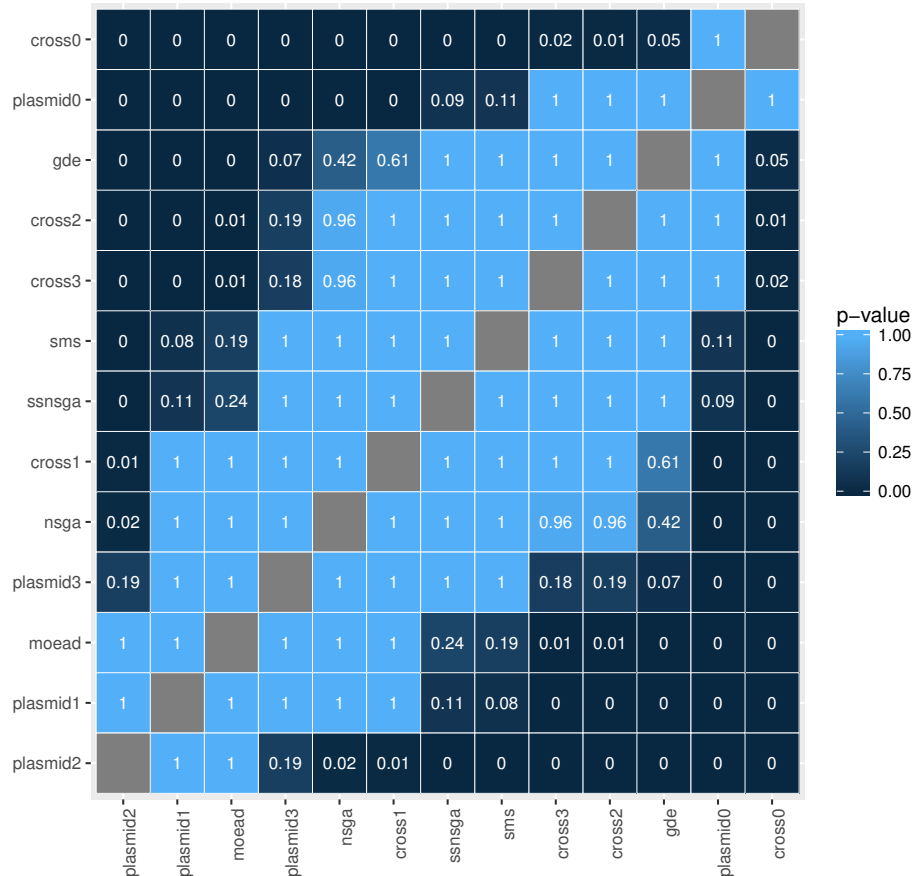


Tabela 67 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.139	0.283	0.286	0.278	0.245	0.334	0.294	0.244	0.303	0.409	0.312	0.277	0.279
2	0.153	0.349	0.318	0.285	0.271	0.319	0.286	0.251	0.332	0.414	0.347	0.271	0.262
3	0.143	0.286	0.262	0.294	0.293	0.264	0.252	0.197	0.304	0.369	0.327	0.228	0.234
4	0.132	0.242	0.289	0.256	0.232	0.282	0.261	0.220	0.296	0.411	0.296	0.245	0.249
5	0.118	0.238	0.344	0.260	0.200	0.253	0.229	0.184	0.245	0.425	0.307	0.212	0.211
6	0.229	0.316	0.221	0.245	0.316	0.372	0.350	0.282	0.365	0.344	0.256	0.340	0.342
7	0.302	0.453	0.242	0.238	0.412	0.490	0.465	0.399	0.521	0.387	0.289	0.461	0.461
8	0.152	0.293	0.256	0.303	0.261	0.326	0.316	0.262	0.354	0.437	0.369	0.290	0.297
9	0.146	0.270	0.306	0.312	0.277	0.310	0.262	0.220	0.302	0.423	0.377	0.257	0.253
10	0.167	0.307	0.301	0.288	0.320	0.314	0.295	0.227	0.308	0.400	0.334	0.284	0.286
11	0.181	0.300	0.284	0.294	0.311	0.346	0.334	0.281	0.352	0.417	0.368	0.311	0.316
12	0.149	0.306	0.273	0.292	0.240	0.308	0.295	0.232	0.332	0.425	0.314	0.282	0.284
13	0.179	0.323	0.242	0.171	0.285	0.399	0.375	0.291	0.402	0.354	0.256	0.359	0.362
14	0.221	0.344	0.235	0.256	0.306	0.389	0.377	0.287	0.403	0.356	0.303	0.356	0.347
15	0.148	0.293	0.267	0.315	0.290	0.305	0.279	0.212	0.279	0.382	0.326	0.262	0.263
16	0.164	0.305	0.241	0.276	0.267	0.356	0.352	0.246	0.348	0.360	0.275	0.345	0.334
17	0.138	0.318	0.288	0.251	0.231	0.383	0.341	0.274	0.415	0.420	0.320	0.331	0.334
18	0.159	0.322	0.258	0.256	0.259	0.315	0.294	0.229	0.374	0.327	0.292	0.275	0.278
19	0.142	0.283	0.304	0.283	0.214	0.311	0.275	0.238	0.297	0.453	0.338	0.269	0.264
20	0.166	0.346	0.295	0.292	0.301	0.348	0.312	0.282	0.376	0.436	0.381	0.303	0.305

Figura 59 – *Posthoc* de Shaffer para I_H no conjunto com $D^k = 20$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 68 – Medianas do indicador de qualidade I_ϵ no conjunto com $D^k=20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.084	1.206	1.200	1.164	1.155	1.220	1.190	1.190	1.190	1.236	1.182	1.181	1.180
2	1.093	1.236	1.286	1.196	1.166	1.190	1.190	1.190	1.238	1.286	1.232	1.147	1.159
3	1.100	1.215	1.200	1.161	1.200	1.149	1.148	1.148	1.204	1.222	1.178	1.133	1.143
4	1.098	1.173	1.195	1.132	1.156	1.233	1.172	1.172	1.214	1.244	1.146	1.172	1.171
5	1.069	1.157	1.330	1.227	1.117	1.167	1.151	1.138	1.154	1.352	1.227	1.136	1.130
6	1.132	1.156	1.171	1.168	1.179	1.214	1.190	1.167	1.222	1.237	1.158	1.179	1.179
7	1.196	1.196	1.118	1.167	1.235	1.286	1.250	1.224	1.286	1.198	1.182	1.261	1.255
8	1.092	1.179	1.158	1.167	1.177	1.233	1.205	1.190	1.238	1.263	1.193	1.188	1.186
9	1.083	1.198	1.271	1.229	1.160	1.192	1.167	1.179	1.222	1.333	1.271	1.167	1.168
10	1.092	1.214	1.240	1.180	1.203	1.200	1.174	1.174	1.217	1.280	1.220	1.174	1.174
11	1.107	1.217	1.220	1.195	1.201	1.220	1.216	1.217	1.261	1.268	1.220	1.196	1.188
12	1.094	1.225	1.222	1.200	1.173	1.217	1.182	1.182	1.227	1.311	1.222	1.184	1.182
13	1.109	1.136	1.144	1.122	1.166	1.304	1.267	1.227	1.227	1.207	1.182	1.261	1.237
14	1.122	1.206	1.135	1.171	1.164	1.263	1.226	1.207	1.250	1.213	1.162	1.226	1.220
15	1.090	1.213	1.180	1.190	1.169	1.179	1.175	1.143	1.179	1.240	1.200	1.143	1.153
16	1.089	1.163	1.125	1.159	1.167	1.208	1.195	1.180	1.227	1.188	1.154	1.188	1.182
17	1.091	1.200	1.158	1.158	1.135	1.286	1.244	1.200	1.316	1.235	1.161	1.229	1.241
18	1.095	1.233	1.167	1.179	1.168	1.196	1.160	1.160	1.250	1.185	1.160	1.148	1.148
19	1.096	1.207	1.217	1.181	1.151	1.214	1.174	1.174	1.175	1.304	1.174	1.174	1.171
20	1.095	1.222	1.231	1.173	1.190	1.217	1.182	1.189	1.286	1.269	1.231	1.182	1.182

Figura 60 – *Posthoc* de Shaffer para I_ϵ no conjunto com $D^k=20$.

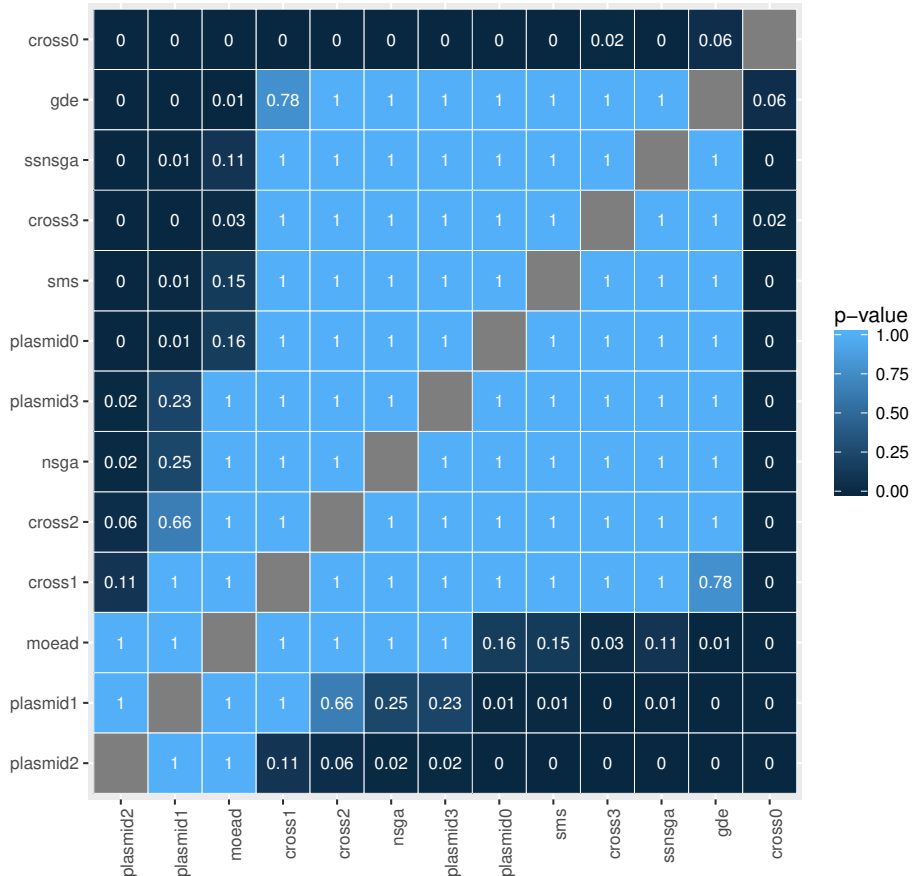
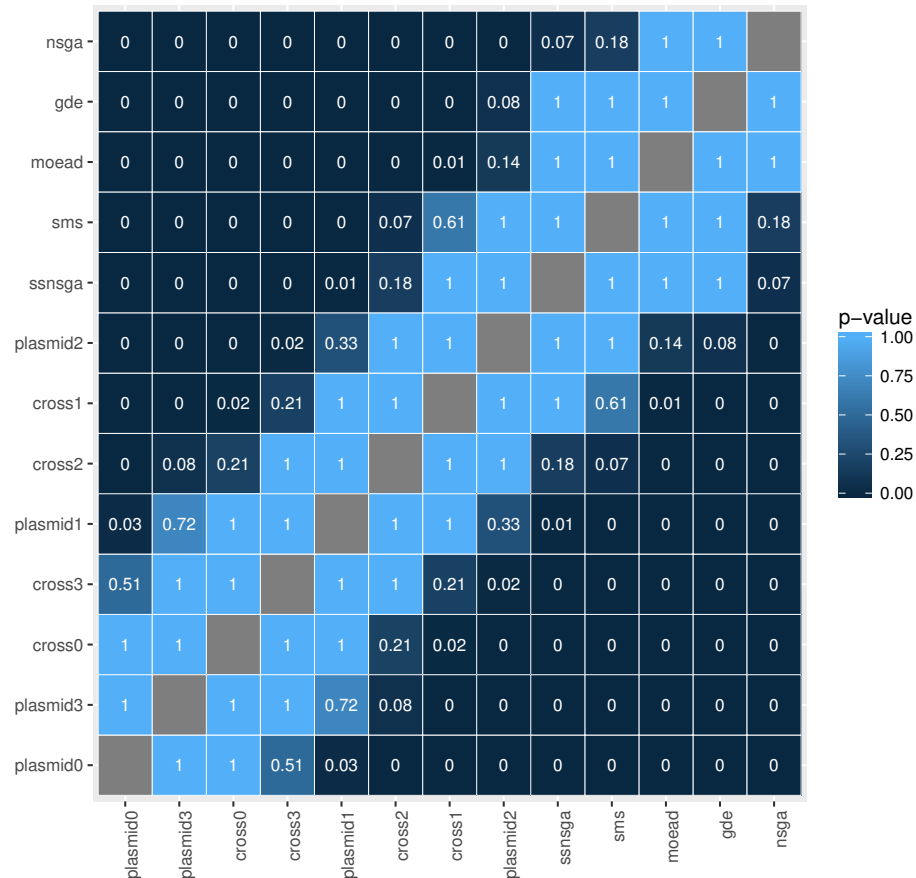


Tabela 69 – Medianas do tempo de processamento no conjunto com $D^k = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	5.476	3.883	4.500	5.285	1.641	1.857	0.852	6.600	4.947	3.273	5.689	2.584	2.628
2	5.586	3.849	4.622	5.302	1.614	1.769	0.820	6.633	4.856	3.343	5.688	2.601	2.639
3	5.343	3.737	4.383	5.153	1.674	1.711	0.786	6.306	4.731	3.198	5.500	2.563	2.592
4	5.633	3.812	4.710	5.549	1.682	1.884	0.869	6.755	4.856	3.362	5.743	2.628	2.672
5	5.258	3.688	4.307	5.120	1.606	1.679	0.724	6.101	4.591	2.886	5.078	2.442	2.492
6	5.669	3.866	4.753	5.573	1.582	1.949	0.910	7.173	5.053	3.620	5.357	2.638	2.695
7	5.570	3.803	4.663	5.605	1.712	1.833	0.826	6.758	4.803	3.270	6.038	2.576	2.592
8	5.472	3.797	4.490	5.324	1.717	1.801	0.823	6.579	4.799	3.295	5.745	2.568	2.610
9	5.660	3.903	4.693	5.444	1.873	1.779	0.857	6.683	4.799	3.407	5.671	2.642	2.685
10	5.550	3.827	4.594	5.402	1.857	1.842	0.844	6.534	4.830	3.231	5.603	2.611	2.657
11	5.562	3.888	4.567	5.314	2.014	1.763	0.827	6.693	4.877	3.516	5.351	2.608	2.649
12	5.441	3.752	4.537	5.280	1.835	1.700	0.760	6.347	4.830	3.232	5.260	2.511	2.536
13	5.557	3.800	4.612	5.462	1.939	1.917	0.870	6.866	4.877	3.398	5.775	2.605	2.628
14	5.490	3.650	4.686	5.183	1.916	1.862	0.842	6.658	4.785	3.261	5.520	2.585	2.580
15	5.644	3.941	4.673	5.552	1.994	1.980	0.937	6.971	4.995	3.580	5.987	2.688	2.729
16	5.578	3.841	4.632	5.567	1.797	1.874	0.850	6.811	4.898	3.413	5.765	2.583	2.615
17	5.553	3.742	4.627	5.412	1.721	1.841	0.817	6.639	4.812	3.274	5.535	2.553	2.576
18	5.553	3.829	4.632	5.378	1.859	1.861	0.854	6.768	4.842	3.496	5.796	2.592	2.641
19	5.647	3.799	4.731	5.408	1.859	1.896	0.851	6.756	4.831	3.443	5.713	2.593	2.625
20	5.790	3.919	4.755	5.548	1.863	1.846	0.877	6.900	4.864	3.508	5.989	2.659	2.694

Figura 61 – *Posthoc* de Shaffer para o Tempo no conjunto com $D^k = 20$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 70 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $D^k = 30$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.114	0.284	0.295	0.227	0.190	0.318	0.248	0.227	0.273	0.364	0.295	0.227	0.227
2	0.271	0.271	0.167	0.267	0.271	0.320	0.271	0.271	0.280	0.271	0.200	0.271	0.271
3	0.122	0.345	0.214	0.320	0.233	0.240	0.204	0.202	0.280	0.286	0.204	0.200	0.200
4	0.108	0.228	0.238	0.188	0.213	0.279	0.227	0.227	0.273	0.262	0.190	0.227	0.218
5	0.125	0.251	0.192	0.220	0.226	0.288	0.231	0.224	0.333	0.260	0.192	0.212	0.212
6	0.154	0.284	0.308	0.245	0.227	0.308	0.240	0.240	0.280	0.385	0.269	0.240	0.240
7	0.106	0.292	0.200	0.231	0.197	0.240	0.220	0.238	0.320	0.260	0.213	0.195	0.190
8	0.114	0.279	0.143	0.256	0.198	0.231	0.192	0.231	0.235	0.231	0.192	0.192	0.185
9	0.143	0.301	0.204	0.246	0.212	0.286	0.245	0.265	0.333	0.286	0.238	0.238	0.245
10	0.116	0.284	0.273	0.239	0.197	0.245	0.217	0.260	0.348	0.364	0.255	0.200	0.217
11	0.111	0.272	0.200	0.230	0.221	0.250	0.247	0.250	0.317	0.289	0.228	0.208	0.209
12	0.438	0.861	0.176	0.191	0.431	0.490	0.424	0.431	0.861	0.255	0.188	0.431	0.431
13	0.125	0.282	0.157	0.207	0.180	0.293	0.237	0.248	0.285	0.216	0.208	0.235	0.216
14	0.420	0.862	0.124	0.213	0.306	0.491	0.401	0.431	0.854	0.217	0.174	0.402	0.422
15	0.136	0.231	0.109	0.208	0.186	0.337	0.304	0.273	0.275	0.273	0.217	0.302	0.273
16	0.143	0.264	0.238	0.211	0.203	0.310	0.240	0.240	0.320	0.298	0.190	0.240	0.239
17	0.111	0.268	0.207	0.190	0.209	0.224	0.192	0.219	0.308	0.259	0.190	0.192	0.192
18	0.118	0.274	0.196	0.220	0.196	0.275	0.238	0.238	0.333	0.286	0.216	0.238	0.238
19	0.125	0.281	0.135	0.255	0.184	0.292	0.240	0.228	0.254	0.250	0.191	0.216	0.216
20	0.125	0.288	0.225	0.233	0.183	0.300	0.250	0.250	0.400	0.325	0.246	0.250	0.250

Figura 62 – *Posthoc* de Shaffer para I_{ϵ_+} no conjunto com $D^k = 30$.

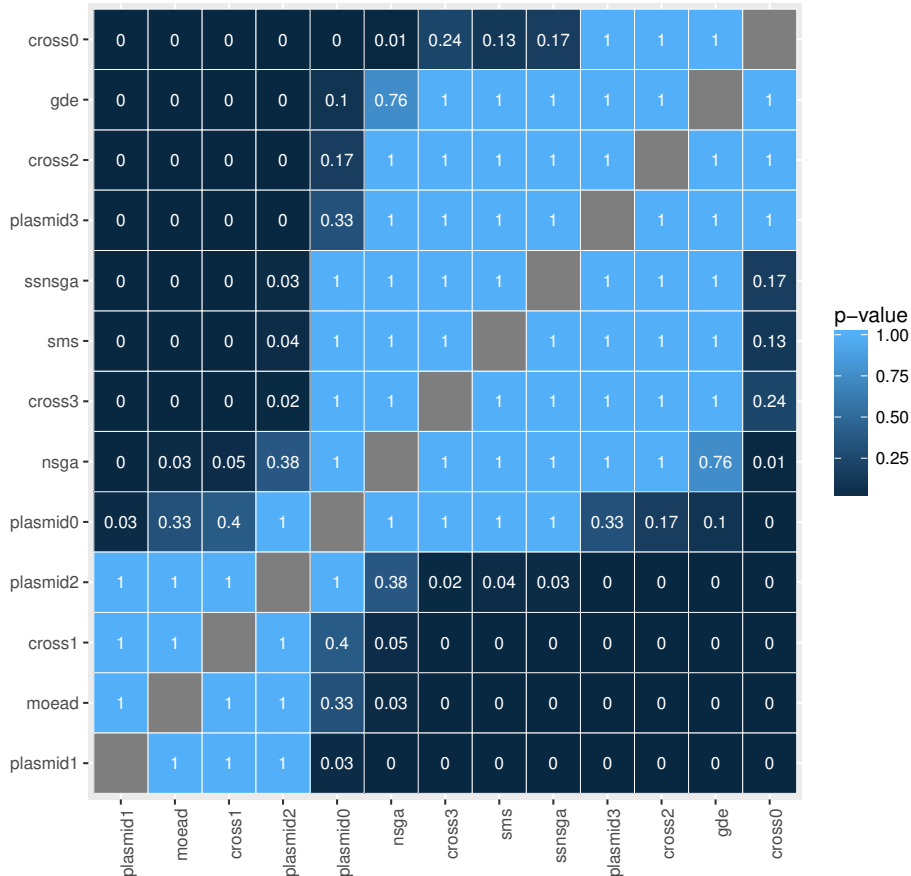
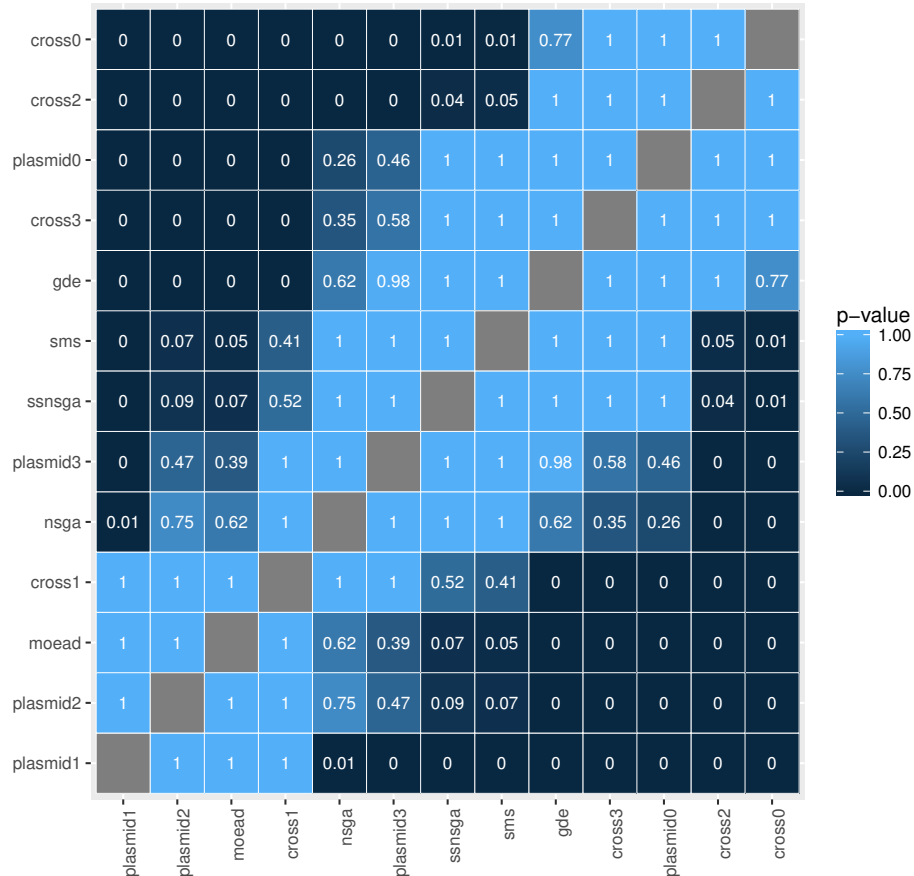


Tabela 71 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 30$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.200	0.353	0.328	0.345	0.320	0.390	0.334	0.300	0.378	0.471	0.415	0.325	0.316
2	0.362	0.523	0.280	0.327	0.451	0.547	0.527	0.468	0.591	0.426	0.370	0.501	0.507
3	0.194	0.413	0.291	0.409	0.316	0.363	0.338	0.282	0.418	0.415	0.386	0.327	0.321
4	0.180	0.307	0.265	0.237	0.299	0.382	0.335	0.284	0.398	0.392	0.316	0.319	0.323
5	0.198	0.380	0.287	0.322	0.333	0.372	0.341	0.304	0.447	0.407	0.362	0.330	0.333
6	0.215	0.346	0.327	0.345	0.362	0.408	0.337	0.322	0.370	0.463	0.384	0.326	0.328
7	0.196	0.396	0.323	0.307	0.300	0.371	0.343	0.295	0.446	0.413	0.339	0.328	0.327
8	0.167	0.338	0.210	0.282	0.285	0.309	0.288	0.269	0.408	0.359	0.323	0.262	0.269
9	0.209	0.377	0.291	0.337	0.320	0.387	0.349	0.295	0.441	0.441	0.381	0.340	0.337
10	0.181	0.342	0.297	0.343	0.304	0.361	0.321	0.306	0.424	0.481	0.399	0.303	0.307
11	0.160	0.363	0.290	0.350	0.298	0.348	0.322	0.290	0.443	0.451	0.424	0.307	0.312
12	0.662	1.036	0.240	0.239	0.651	0.764	0.642	0.657	1.032	0.377	0.285	0.658	0.659
13	0.203	0.396	0.245	0.274	0.292	0.410	0.342	0.301	0.434	0.362	0.312	0.334	0.337
14	0.602	1.127	0.238	0.255	0.467	0.751	0.618	0.586	1.129	0.362	0.272	0.610	0.628
15	0.253	0.402	0.216	0.230	0.337	0.452	0.407	0.366	0.528	0.453	0.345	0.398	0.398
16	0.220	0.350	0.322	0.279	0.324	0.411	0.360	0.311	0.450	0.454	0.338	0.344	0.348
17	0.167	0.390	0.274	0.307	0.306	0.335	0.299	0.255	0.415	0.372	0.334	0.291	0.296
18	0.182	0.344	0.293	0.287	0.308	0.381	0.341	0.300	0.445	0.455	0.327	0.329	0.333
19	0.209	0.375	0.260	0.302	0.301	0.417	0.359	0.317	0.438	0.417	0.334	0.344	0.346
20	0.206	0.379	0.323	0.360	0.318	0.425	0.373	0.333	0.516	0.502	0.418	0.366	0.356

Figura 63 – *Posthoc* de Shaffer para I_H no conjunto com $D^k = 30$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 72 – Medianas do indicador de qualidade I_ϵ no conjunto com $D^k = 30$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.108	1.284	1.295	1.227	1.189	1.261	1.217	1.217	1.273	1.364	1.295	1.217	1.192
2	1.271	1.271	1.167	1.267	1.271	1.300	1.271	1.271	1.280	1.260	1.186	1.271	1.271
3	1.122	1.345	1.214	1.320	1.233	1.216	1.192	1.192	1.280	1.286	1.204	1.185	1.179
4	1.107	1.228	1.238	1.188	1.213	1.261	1.217	1.217	1.273	1.262	1.182	1.217	1.198
5	1.115	1.251	1.192	1.220	1.226	1.250	1.200	1.200	1.333	1.250	1.186	1.200	1.200
6	1.131	1.284	1.308	1.245	1.225	1.275	1.215	1.224	1.280	1.385	1.269	1.202	1.200
7	1.104	1.292	1.200	1.231	1.197	1.240	1.190	1.208	1.310	1.245	1.186	1.190	1.190
8	1.113	1.279	1.143	1.256	1.198	1.231	1.192	1.214	1.231	1.231	1.192	1.175	1.173
9	1.124	1.300	1.204	1.246	1.212	1.268	1.227	1.227	1.333	1.286	1.224	1.227	1.214
10	1.112	1.284	1.273	1.239	1.197	1.217	1.204	1.212	1.348	1.364	1.255	1.174	1.177
11	1.107	1.272	1.200	1.230	1.221	1.234	1.209	1.235	1.313	1.289	1.222	1.208	1.208
12	1.386	1.861	1.176	1.191	1.382	1.444	1.373	1.383	1.861	1.255	1.169	1.379	1.383
13	1.120	1.282	1.157	1.207	1.176	1.275	1.210	1.212	1.266	1.216	1.186	1.208	1.203
14	1.400	1.862	1.121	1.213	1.284	1.463	1.374	1.411	1.854	1.217	1.158	1.364	1.387
15	1.136	1.231	1.109	1.208	1.185	1.330	1.292	1.273	1.273	1.261	1.208	1.273	1.273
16	1.137	1.264	1.238	1.211	1.203	1.282	1.222	1.222	1.320	1.298	1.173	1.227	1.222
17	1.110	1.268	1.207	1.190	1.209	1.193	1.192	1.192	1.308	1.259	1.190	1.167	1.185
18	1.118	1.274	1.196	1.220	1.196	1.258	1.227	1.237	1.333	1.275	1.190	1.218	1.218
19	1.119	1.281	1.135	1.255	1.184	1.280	1.213	1.214	1.250	1.243	1.176	1.216	1.211
20	1.116	1.288	1.225	1.233	1.179	1.267	1.228	1.238	1.400	1.325	1.225	1.227	1.222

Figura 64 – *Posthoc* de Shaffer para I_ϵ no conjunto com $D^k = 30$.

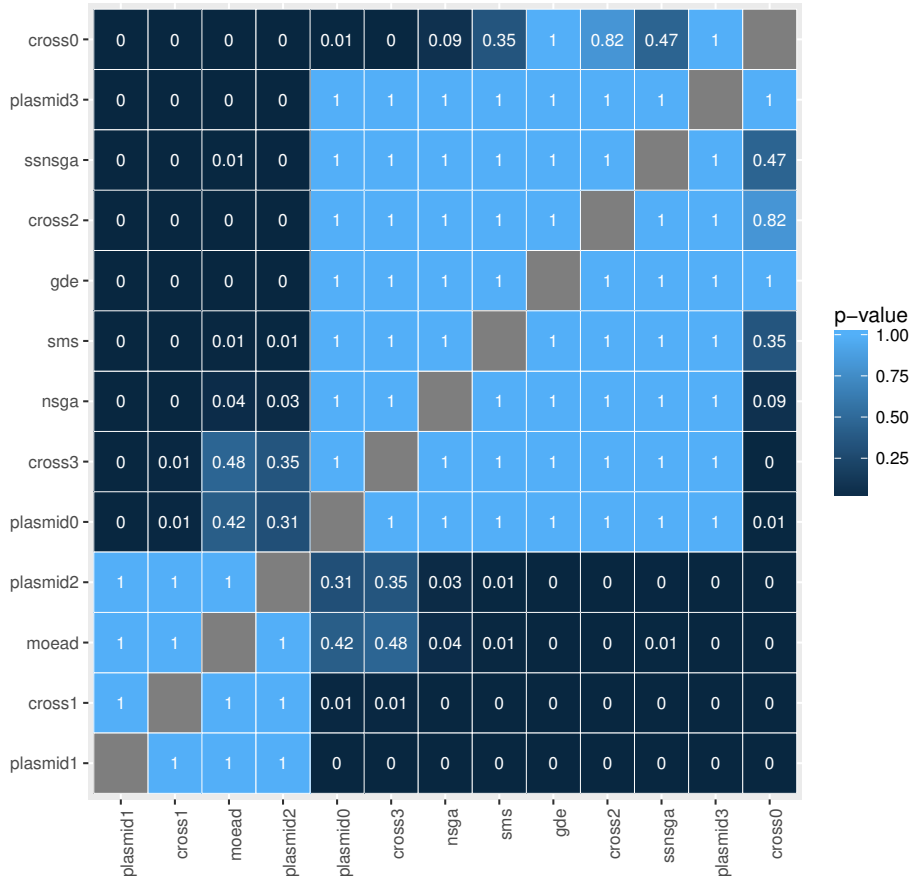
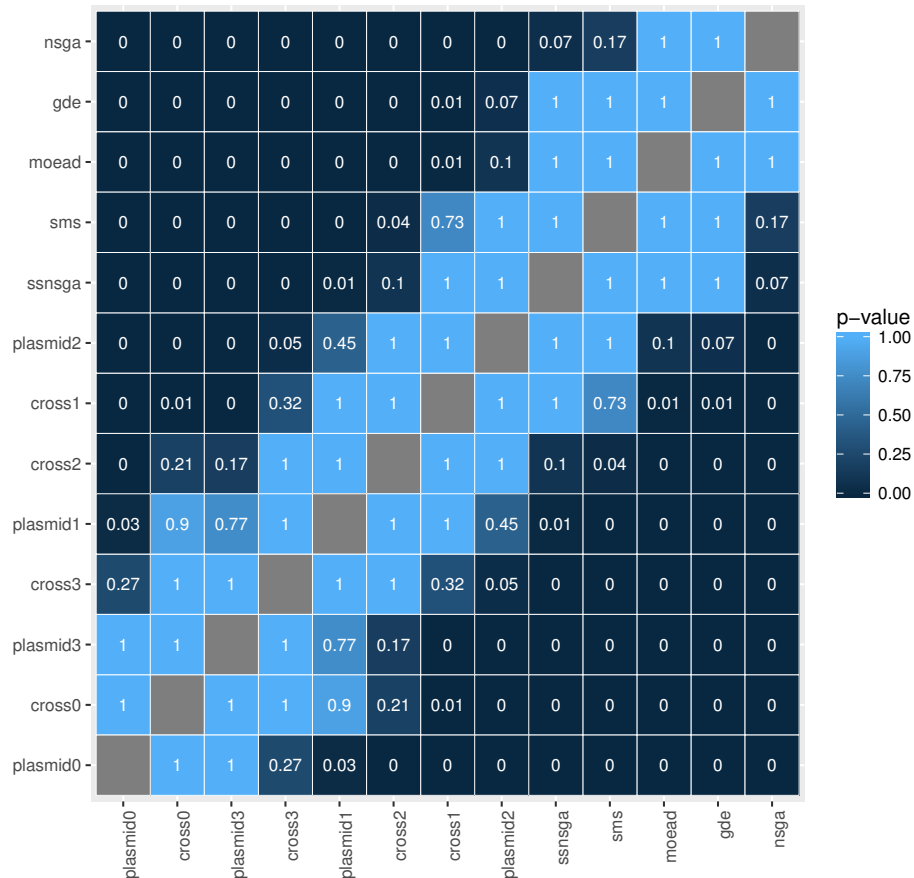


Tabela 73 – Medianas do tempo de processamento no conjunto com $D^k = 30$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	8.992	5.952	7.499	8.481	2.664	2.872	1.405	11.085	7.478	5.819	9.346	4.109	4.172
2	8.532	5.631	6.950	8.087	2.507	2.726	1.358	11.070	7.263	5.706	8.841	3.997	4.091
3	9.291	5.993	8.127	8.316	2.719	2.990	1.425	11.488	7.647	6.306	9.540	4.170	4.215
4	8.314	5.699	7.093	7.993	2.440	2.674	1.300	10.539	7.268	5.393	8.690	3.911	3.990
5	9.004	5.925	7.465	8.495	2.706	2.737	1.344	11.126	7.379	6.031	9.109	4.086	4.166
6	8.701	5.818	7.033	8.144	2.749	2.572	1.288	10.632	7.251	5.352	8.700	4.021	4.120
7	8.640	5.801	7.153	8.093	2.712	2.756	1.317	10.801	7.498	5.556	9.029	3.963	4.056
8	8.140	5.437	6.723	7.419	2.757	2.544	1.224	10.484	7.184	5.262	8.475	3.906	3.975
9	9.157	5.971	7.876	8.426	2.796	2.907	1.398	11.032	7.629	5.815	9.285	4.097	4.191
10	9.168	6.122	7.489	8.234	2.834	2.901	1.439	11.510	7.742	5.928	9.027	4.145	4.251
11	8.974	6.051	7.476	8.394	2.850	2.663	1.304	10.998	7.498	5.727	8.818	4.072	4.165
12	9.640	6.965	7.231	8.332	2.612	2.762	1.537	11.720	9.163	5.700	9.113	4.231	4.344
13	8.666	5.780	7.118	8.205	2.642	2.613	1.249	10.322	7.343	5.273	8.945	3.976	4.070
14	9.081	6.908	6.979	8.113	2.695	2.603	1.432	10.925	8.876	5.241	8.723	4.153	4.251
15	9.142	5.956	7.664	8.256	2.729	2.839	1.381	11.176	7.591	5.854	8.748	4.112	4.183
16	8.885	5.882	7.334	8.463	2.811	2.644	1.281	10.642	7.274	5.568	8.569	4.001	4.074
17	8.456	5.882	6.977	8.252	2.767	2.558	1.263	10.515	7.199	5.455	8.248	3.984	4.092
18	8.694	5.757	7.146	8.121	2.768	2.830	1.347	10.812	7.433	5.587	8.914	4.009	4.089
19	8.865	5.949	7.342	8.100	2.505	2.904	1.416	11.005	7.562	5.548	9.216	4.085	4.153
20	8.742	5.757	7.243	8.094	2.706	2.642	1.280	10.665	7.265	5.536	8.685	4.001	4.102

Figura 65 – *Posthoc* de Shaffer para o Tempo no conjunto com $D^k = 30$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 74 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $D^k = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.222	0.312	0.118	0.284	0.229	0.289	0.267	0.272	0.320	0.213	0.240	0.267	0.267
2	0.131	0.343	0.136	0.317	0.199	0.322	0.273	0.273	0.318	0.304	0.182	0.261	0.239
3	0.156	0.351	0.265	0.269	0.226	0.320	0.242	0.265	0.292	0.347	0.240	0.240	0.224
4	0.182	0.308	0.133	0.258	0.227	0.409	0.333	0.318	0.364	0.227	0.227	0.332	0.318
5	0.123	0.349	0.276	0.308	0.187	0.286	0.231	0.265	0.273	0.408	0.229	0.231	0.224
6	0.136	0.347	0.226	0.292	0.234	0.283	0.227	0.264	0.364	0.302	0.236	0.226	0.225
7	0.160	0.373	0.294	0.278	0.216	0.360	0.280	0.294	0.360	0.402	0.202	0.280	0.275
8	0.125	0.267	0.152	0.269	0.214	0.219	0.206	0.217	0.250	0.283	0.246	0.196	0.196
9	0.138	0.361	0.300	0.318	0.220	0.286	0.241	0.267	0.381	0.390	0.280	0.240	0.238
10	0.140	0.330	0.205	0.308	0.239	0.318	0.250	0.292	0.333	0.352	0.250	0.250	0.250
11	0.134	0.325	0.268	0.318	0.227	0.261	0.217	0.239	0.304	0.341	0.268	0.217	0.217
12	0.141	0.353	0.277	0.298	0.254	0.261	0.217	0.230	0.348	0.362	0.319	0.186	0.190
13	0.143	0.341	0.275	0.338	0.243	0.235	0.208	0.234	0.308	0.353	0.294	0.208	0.208
14	0.129	0.312	0.313	0.268	0.199	0.273	0.227	0.246	0.318	0.354	0.250	0.227	0.227
15	0.125	0.348	0.263	0.284	0.211	0.276	0.211	0.211	0.333	0.342	0.211	0.224	0.208
16	0.126	0.299	0.163	0.263	0.199	0.265	0.227	0.250	0.286	0.224	0.179	0.224	0.224
17	0.511	0.912	0.126	0.350	0.494	0.581	0.511	0.511	0.909	0.242	0.208	0.490	0.471
18	0.138	0.352	0.220	0.326	0.218	0.240	0.220	0.226	0.320	0.280	0.240	0.200	0.200
19	0.297	0.323	0.243	0.279	0.297	0.407	0.318	0.318	0.324	0.371	0.270	0.324	0.324
20	0.567	0.889	0.262	0.295	0.580	0.648	0.571	0.576	0.880	0.333	0.192	0.576	0.558

Figura 66 – *Posthoc* de Shaffer para $I_{\epsilon+}$ no conjunto com $D^k = 40$.

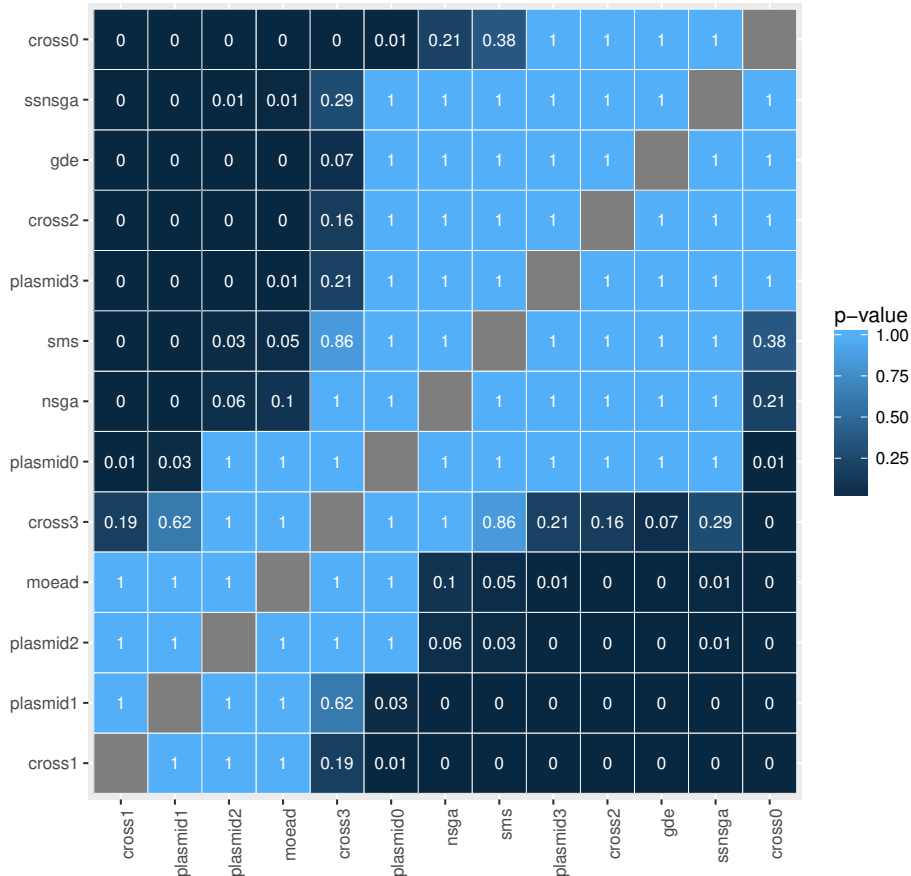
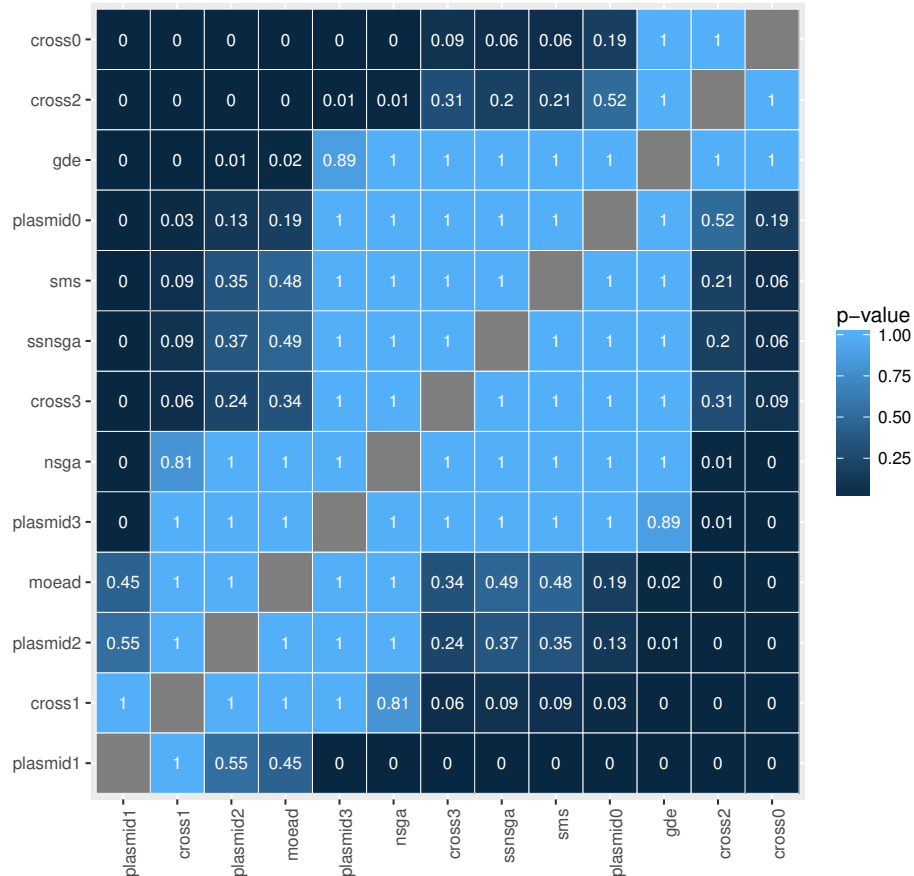


Tabela 75 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.350	0.555	0.211	0.352	0.443	0.492	0.481	0.428	0.630	0.355	0.406	0.461	0.475
2	0.251	0.492	0.262	0.345	0.352	0.455	0.421	0.399	0.560	0.463	0.319	0.396	0.404
3	0.253	0.418	0.338	0.374	0.368	0.445	0.392	0.364	0.458	0.451	0.410	0.370	0.371
4	0.335	0.521	0.243	0.321	0.427	0.608	0.537	0.488	0.649	0.398	0.375	0.520	0.525
5	0.166	0.384	0.285	0.350	0.262	0.345	0.312	0.317	0.419	0.495	0.380	0.298	0.302
6	0.203	0.428	0.267	0.359	0.317	0.383	0.336	0.320	0.508	0.428	0.417	0.324	0.322
7	0.239	0.425	0.346	0.327	0.360	0.475	0.406	0.406	0.495	0.517	0.375	0.398	0.401
8	0.170	0.330	0.230	0.331	0.279	0.330	0.320	0.276	0.384	0.411	0.374	0.304	0.303
9	0.216	0.454	0.361	0.397	0.313	0.408	0.378	0.366	0.503	0.517	0.463	0.370	0.371
10	0.210	0.374	0.264	0.364	0.332	0.410	0.371	0.355	0.474	0.493	0.428	0.347	0.359
11	0.176	0.357	0.322	0.433	0.309	0.356	0.327	0.290	0.406	0.451	0.436	0.314	0.309
12	0.208	0.396	0.341	0.431	0.342	0.365	0.339	0.315	0.472	0.451	0.474	0.309	0.317
13	0.193	0.404	0.316	0.429	0.342	0.335	0.322	0.294	0.438	0.452	0.445	0.315	0.309
14	0.184	0.364	0.351	0.402	0.277	0.350	0.325	0.301	0.458	0.468	0.439	0.313	0.316
15	0.172	0.405	0.324	0.375	0.296	0.342	0.318	0.282	0.477	0.468	0.402	0.302	0.297
16	0.216	0.395	0.241	0.311	0.307	0.385	0.349	0.329	0.475	0.355	0.314	0.340	0.341
17	0.791	1.151	0.250	0.376	0.769	0.871	0.762	0.808	1.145	0.386	0.372	0.762	0.752
18	0.215	0.439	0.289	0.412	0.315	0.375	0.344	0.310	0.489	0.407	0.428	0.325	0.330
19	0.374	0.555	0.319	0.347	0.444	0.629	0.544	0.498	0.639	0.522	0.434	0.529	0.533
20	0.786	1.089	0.360	0.344	0.812	0.907	0.818	0.811	1.088	0.487	0.347	0.815	0.772

Figura 67 – *Posthoc* de Shaffer para I_H no conjunto com $D^k = 40$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 76 – Medianas do indicador de qualidade I_ϵ no conjunto com $D^k=40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.222	1.312	1.118	1.284	1.228	1.286	1.247	1.256	1.320	1.202	1.221	1.244	1.244
2	1.126	1.343	1.136	1.317	1.199	1.307	1.235	1.254	1.318	1.287	1.158	1.234	1.222
3	1.145	1.351	1.265	1.269	1.224	1.296	1.230	1.240	1.280	1.347	1.222	1.200	1.200
4	1.179	1.308	1.133	1.258	1.214	1.396	1.313	1.304	1.364	1.222	1.217	1.315	1.308
5	1.119	1.349	1.276	1.308	1.187	1.269	1.209	1.231	1.269	1.408	1.224	1.222	1.197
6	1.121	1.347	1.226	1.292	1.234	1.268	1.227	1.228	1.333	1.291	1.226	1.204	1.200
7	1.143	1.373	1.294	1.278	1.210	1.333	1.244	1.265	1.345	1.402	1.196	1.254	1.237
8	1.125	1.267	1.152	1.269	1.214	1.203	1.188	1.200	1.250	1.283	1.223	1.182	1.182
9	1.135	1.361	1.300	1.318	1.220	1.252	1.227	1.230	1.381	1.390	1.280	1.217	1.217
10	1.140	1.330	1.205	1.308	1.239	1.280	1.241	1.250	1.333	1.352	1.250	1.240	1.222
11	1.134	1.325	1.268	1.318	1.227	1.240	1.198	1.217	1.304	1.341	1.268	1.192	1.194
12	1.141	1.353	1.277	1.298	1.254	1.231	1.192	1.214	1.348	1.362	1.319	1.170	1.173
13	1.143	1.341	1.275	1.338	1.243	1.208	1.208	1.208	1.292	1.353	1.294	1.197	1.204
14	1.120	1.312	1.313	1.268	1.199	1.250	1.224	1.227	1.318	1.354	1.250	1.208	1.208
15	1.124	1.348	1.263	1.284	1.211	1.258	1.208	1.208	1.333	1.342	1.211	1.209	1.192
16	1.119	1.299	1.163	1.263	1.196	1.250	1.208	1.226	1.286	1.224	1.167	1.210	1.207
17	1.465	1.912	1.126	1.350	1.449	1.534	1.449	1.471	1.909	1.237	1.204	1.452	1.437
18	1.136	1.352	1.220	1.326	1.218	1.216	1.192	1.200	1.320	1.280	1.220	1.179	1.184
19	1.297	1.323	1.243	1.279	1.297	1.385	1.304	1.301	1.324	1.365	1.243	1.324	1.320
20	1.488	1.889	1.262	1.295	1.502	1.597	1.498	1.500	1.880	1.321	1.184	1.492	1.479

Figura 68 – *Posthoc* de Shaffer para I_ϵ no conjunto com $D^k=40$.

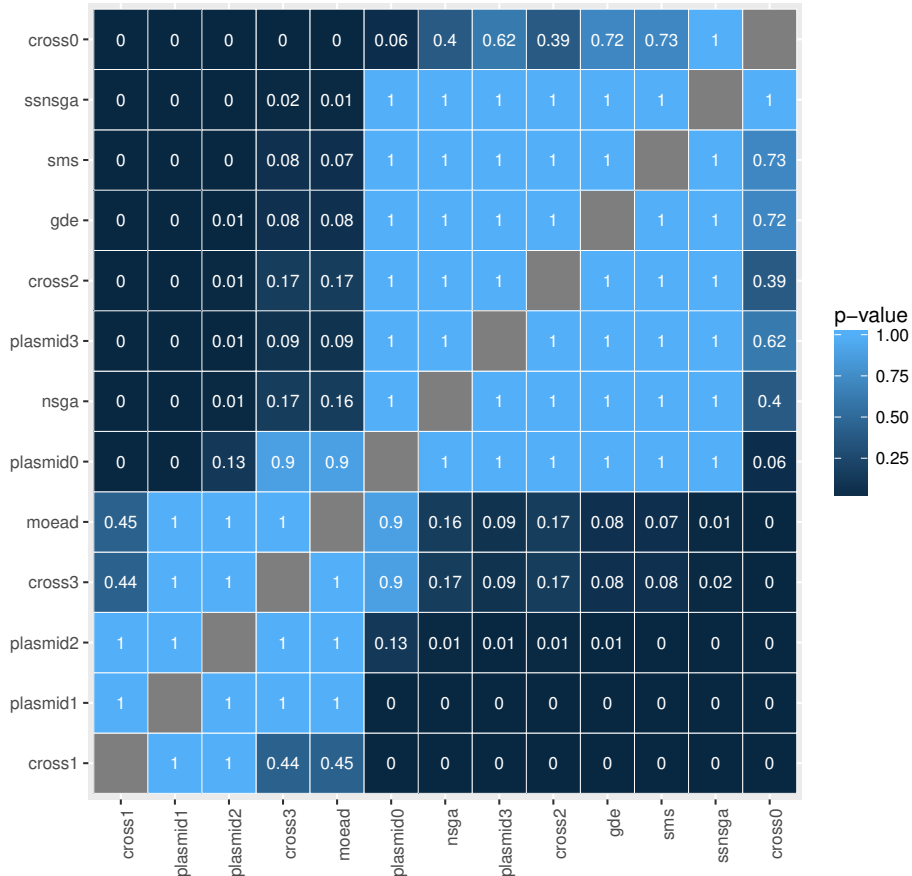
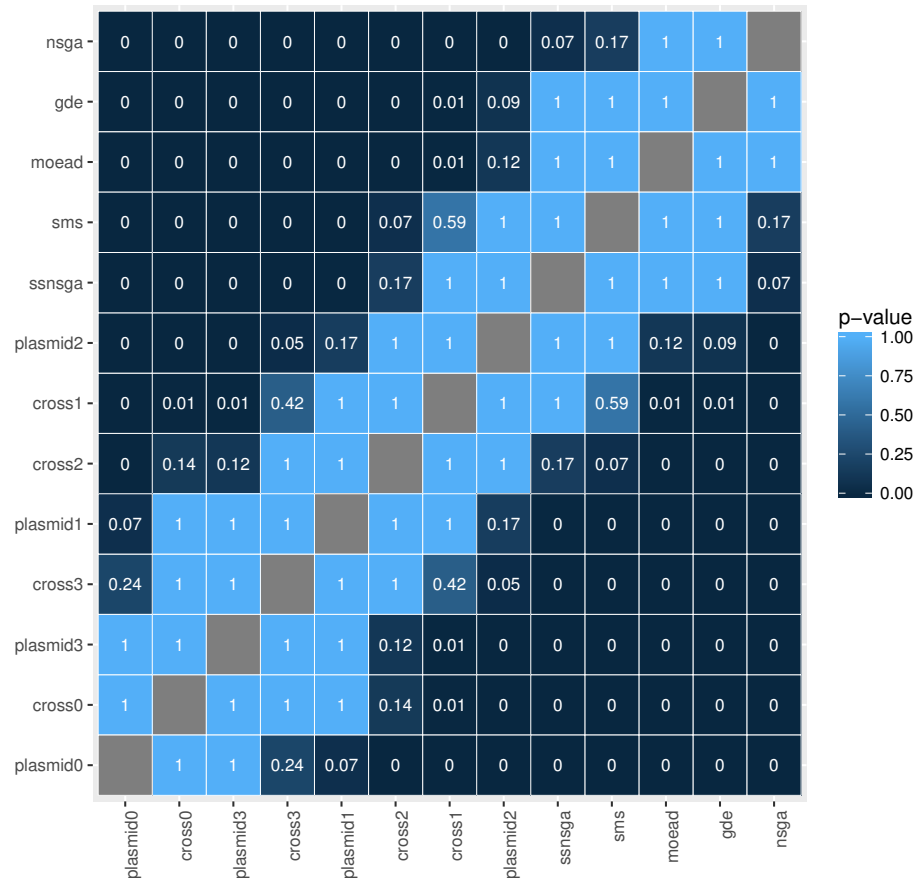


Tabela 77 – Medianas do tempo de processamento no conjunto com $D^k = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	12.521	8.193	10.237	11.559	3.272	3.267	1.737	16.071	11.316	7.967	12.352	5.353	5.458
2	12.411	8.153	10.315	11.249	3.290	3.416	1.653	15.414	11.372	7.642	11.727	5.294	5.380
3	12.398	8.292	10.081	11.719	3.258	3.311	1.643	15.347	10.758	7.413	12.753	5.263	5.330
4	12.021	7.824	10.045	11.349	3.152	3.252	1.656	15.328	10.676	7.809	12.364	5.257	5.325
5	12.437	8.158	10.454	11.562	3.271	3.411	1.640	15.333	11.008	7.738	12.385	5.259	5.337
6	12.442	8.330	10.184	11.670	3.250	3.323	1.676	15.444	10.823	7.842	12.597	5.336	5.385
7	12.389	8.215	10.385	11.870	3.372	3.285	1.633	15.341	10.892	7.679	12.695	5.302	5.386
8	12.622	8.133	10.565	11.961	3.209	3.380	1.660	15.607	10.989	7.788	12.509	5.305	5.381
9	12.104	8.126	10.049	11.701	3.364	3.031	1.477	14.805	10.489	7.278	11.914	5.127	5.248
10	12.122	8.111	9.913	11.454	3.355	3.188	1.592	15.151	10.690	7.704	12.280	5.253	5.332
11	12.491	8.406	10.334	11.827	3.432	3.363	1.688	15.418	10.958	7.750	12.706	5.337	5.395
12	12.327	8.404	10.115	11.455	3.477	3.380	1.692	15.422	10.955	7.807	12.649	5.318	5.423
13	12.770	8.595	10.523	11.732	3.494	3.501	1.783	15.844	11.385	8.031	12.426	5.476	5.561
14	12.387	8.249	10.321	11.779	3.339	3.343	1.681	15.676	10.817	8.031	12.625	5.320	5.428
15	12.512	8.483	10.429	11.532	3.340	3.468	1.700	15.276	11.163	7.573	12.487	5.322	5.438
16	12.339	7.953	10.226	11.472	3.282	3.441	1.709	15.572	10.689	7.963	12.920	5.333	5.397
17	13.684	9.327	10.205	11.416	3.407	3.336	1.829	16.605	12.977	7.533	12.318	5.473	5.540
18	12.792	8.523	10.923	11.975	3.403	3.556	1.766	16.255	11.409	8.072	13.351	5.468	5.528
19	11.743	7.698	9.731	11.409	3.163	3.010	1.489	14.440	10.371	7.150	11.802	5.093	5.175
20	13.922	9.592	10.364	11.464	3.353	3.567	1.985	17.612	13.223	7.960	12.397	5.639	5.693

Figura 69 – *Posthoc* de Shaffer para o Tempo no conjunto com $D^k = 40$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 78 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $D^k = 50$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.146	0.460	0.283	0.389	0.228	0.304	0.229	0.273	0.392	0.413	0.233	0.227	0.227
2	0.486	0.486	0.189	0.324	0.486	0.522	0.486	0.486	0.486	0.297	0.208	0.486	0.486
3	0.143	0.347	0.113	0.328	0.236	0.291	0.250	0.250	0.357	0.205	0.183	0.250	0.250
4	0.571	0.893	0.286	0.348	0.543	0.654	0.570	0.558	0.894	0.457	0.222	0.543	0.563
5	0.166	0.421	0.208	0.328	0.225	0.341	0.255	0.271	0.391	0.271	0.217	0.261	0.250
6	0.161	0.421	0.318	0.358	0.251	0.259	0.205	0.227	0.333	0.432	0.273	0.195	0.185
7	0.114	0.314	0.222	0.270	0.176	0.296	0.245	0.267	0.326	0.311	0.200	0.227	0.222
8	0.378	0.422	0.267	0.374	0.378	0.478	0.356	0.369	0.400	0.411	0.222	0.363	0.356
9	0.140	0.408	0.283	0.316	0.193	0.264	0.217	0.261	0.391	0.358	0.226	0.208	0.217
10	0.142	0.415	0.255	0.382	0.204	0.319	0.255	0.277	0.417	0.340	0.241	0.264	0.234
11	0.154	0.420	0.277	0.342	0.175	0.340	0.261	0.280	0.391	0.362	0.217	0.255	0.234
12	0.545	0.591	0.222	0.395	0.477	0.539	0.500	0.511	0.591	0.295	0.222	0.477	0.477
13	0.170	0.411	0.208	0.365	0.252	0.325	0.257	0.274	0.375	0.264	0.226	0.248	0.247
14	0.146	0.443	0.469	0.361	0.191	0.310	0.250	0.269	0.385	0.542	0.271	0.231	0.231
15	0.129	0.407	0.388	0.298	0.195	0.286	0.208	0.224	0.333	0.449	0.286	0.204	0.184
16	0.193	0.379	0.280	0.328	0.226	0.435	0.317	0.345	0.391	0.439	0.227	0.338	0.301
17	0.153	0.348	0.267	0.327	0.233	0.308	0.261	0.288	0.311	0.333	0.222	0.261	0.222
18	0.588	0.588	0.294	0.355	0.561	0.647	0.588	0.588	0.588	0.441	0.222	0.561	0.567
19	0.170	0.402	0.195	0.356	0.240	0.415	0.308	0.341	0.346	0.317	0.220	0.317	0.308

Figura 70 – *Posthoc* de Shaffer para I_{ϵ_+} no conjunto com $D^k = 50$.

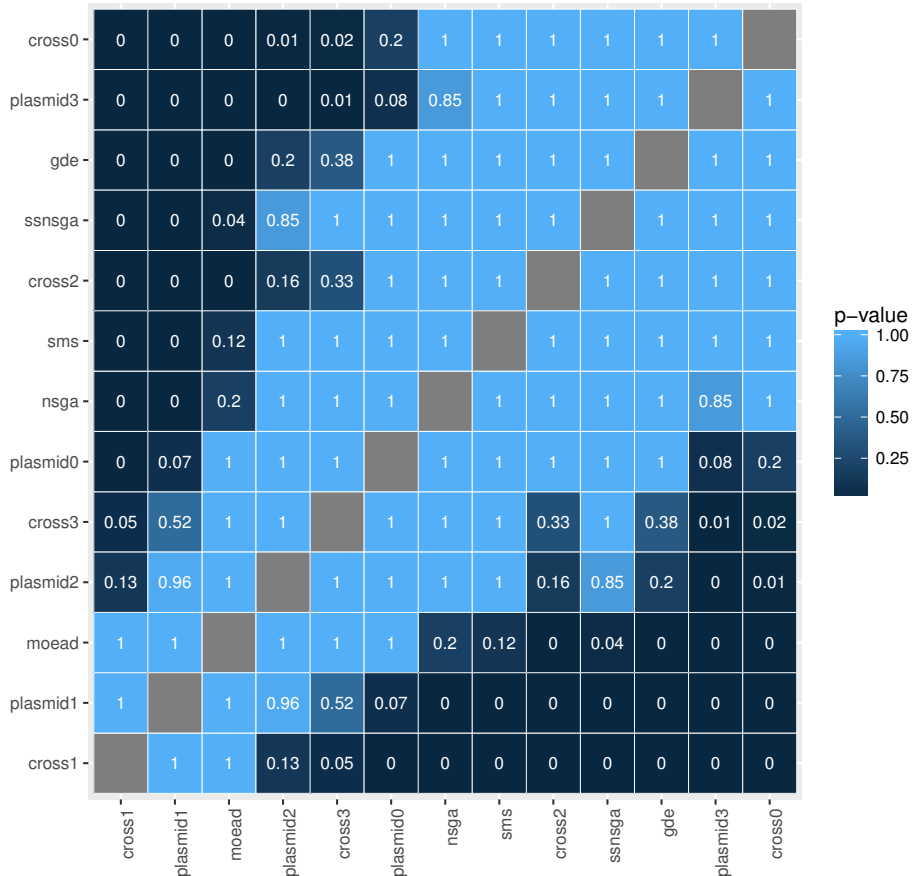
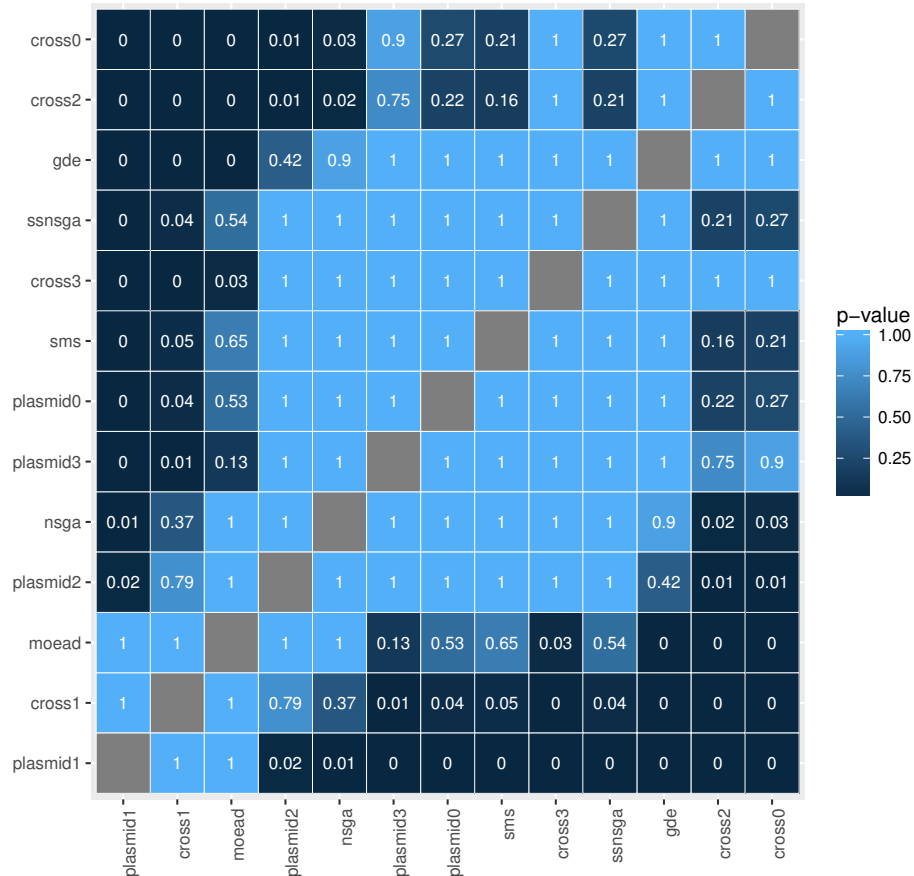


Tabela 79 – Medianas do indicador de qualidade I_H no conjunto com $D^k = 50$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.217	0.476	0.317	0.419	0.292	0.393	0.357	0.360	0.517	0.519	0.424	0.342	0.337
2	0.591	0.757	0.282	0.373	0.635	0.861	0.755	0.710	0.811	0.462	0.378	0.738	0.736
3	0.215	0.455	0.229	0.380	0.340	0.392	0.376	0.322	0.559	0.344	0.369	0.347	0.352
4	0.830	1.060	0.330	0.323	0.791	0.885	0.807	0.824	1.059	0.549	0.376	0.794	0.810
5	0.266	0.499	0.289	0.393	0.350	0.447	0.388	0.366	0.542	0.416	0.410	0.382	0.384
6	0.199	0.441	0.359	0.418	0.320	0.337	0.319	0.296	0.479	0.499	0.422	0.293	0.290
7	0.197	0.423	0.265	0.313	0.279	0.419	0.374	0.351	0.493	0.435	0.355	0.344	0.343
8	0.447	0.681	0.294	0.357	0.491	0.591	0.547	0.514	0.703	0.467	0.366	0.540	0.540
9	0.212	0.458	0.322	0.366	0.273	0.365	0.309	0.297	0.520	0.449	0.367	0.297	0.295
10	0.218	0.475	0.304	0.388	0.304	0.398	0.369	0.360	0.593	0.442	0.406	0.352	0.342
11	0.236	0.479	0.342	0.390	0.303	0.427	0.363	0.346	0.558	0.500	0.417	0.355	0.341
12	0.677	0.848	0.309	0.401	0.675	0.806	0.742	0.717	0.889	0.418	0.395	0.734	0.736
13	0.253	0.513	0.291	0.408	0.351	0.420	0.388	0.369	0.546	0.395	0.413	0.365	0.368
14	0.209	0.444	0.434	0.405	0.289	0.410	0.346	0.342	0.488	0.561	0.408	0.331	0.332
15	0.198	0.438	0.373	0.397	0.291	0.359	0.315	0.312	0.464	0.501	0.426	0.297	0.290
16	0.270	0.420	0.358	0.336	0.349	0.523	0.442	0.430	0.537	0.588	0.403	0.434	0.418
17	0.226	0.391	0.305	0.377	0.322	0.429	0.385	0.375	0.476	0.448	0.391	0.367	0.357
18	0.760	0.843	0.355	0.352	0.824	0.945	0.912	0.912	0.892	0.578	0.376	0.898	0.913
19	0.308	0.530	0.283	0.411	0.393	0.639	0.519	0.477	0.579	0.432	0.403	0.510	0.510

Figura 71 – *Posthoc* de Shaffer para I_H no conjunto com $D^k = 50$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 80 – Medianas do indicador de qualidade I_ϵ no conjunto com $D^k=50$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.142	1.460	1.283	1.389	1.228	1.265	1.210	1.231	1.364	1.413	1.222	1.212	1.194
2	1.459	1.462	1.189	1.324	1.459	1.488	1.459	1.460	1.459	1.297	1.200	1.459	1.459
3	1.143	1.347	1.113	1.328	1.236	1.276	1.241	1.234	1.357	1.200	1.175	1.227	1.231
4	1.484	1.893	1.286	1.348	1.460	1.625	1.475	1.475	1.894	1.457	1.214	1.464	1.470
5	1.150	1.421	1.208	1.328	1.224	1.308	1.226	1.244	1.391	1.271	1.200	1.238	1.226
6	1.161	1.421	1.318	1.358	1.251	1.240	1.185	1.213	1.333	1.432	1.273	1.173	1.173
7	1.114	1.314	1.222	1.270	1.176	1.286	1.223	1.250	1.303	1.311	1.185	1.214	1.204
8	1.347	1.419	1.267	1.374	1.347	1.478	1.319	1.337	1.367	1.411	1.222	1.327	1.319
9	1.134	1.408	1.283	1.316	1.192	1.239	1.191	1.213	1.356	1.358	1.209	1.175	1.178
10	1.140	1.415	1.255	1.382	1.204	1.292	1.240	1.255	1.385	1.340	1.214	1.245	1.222
11	1.137	1.420	1.277	1.342	1.171	1.304	1.222	1.250	1.391	1.362	1.208	1.222	1.215
12	1.545	1.591	1.216	1.395	1.477	1.534	1.500	1.511	1.591	1.289	1.208	1.421	1.422
13	1.164	1.411	1.208	1.365	1.246	1.293	1.241	1.241	1.375	1.264	1.208	1.211	1.224
14	1.133	1.443	1.469	1.361	1.190	1.268	1.214	1.228	1.385	1.542	1.271	1.207	1.197
15	1.129	1.407	1.388	1.298	1.195	1.241	1.184	1.206	1.333	1.449	1.286	1.179	1.163
16	1.174	1.379	1.280	1.328	1.225	1.415	1.304	1.311	1.391	1.439	1.214	1.304	1.273
17	1.149	1.348	1.267	1.327	1.233	1.292	1.247	1.261	1.304	1.333	1.211	1.245	1.217
18	1.588	1.588	1.294	1.355	1.559	1.647	1.588	1.588	1.571	1.441	1.203	1.559	1.559
19	1.155	1.402	1.195	1.356	1.234	1.370	1.274	1.296	1.346	1.317	1.195	1.296	1.279

Figura 72 – *Posthoc* de Shaffer para I_ϵ no conjunto com $D^k=50$.

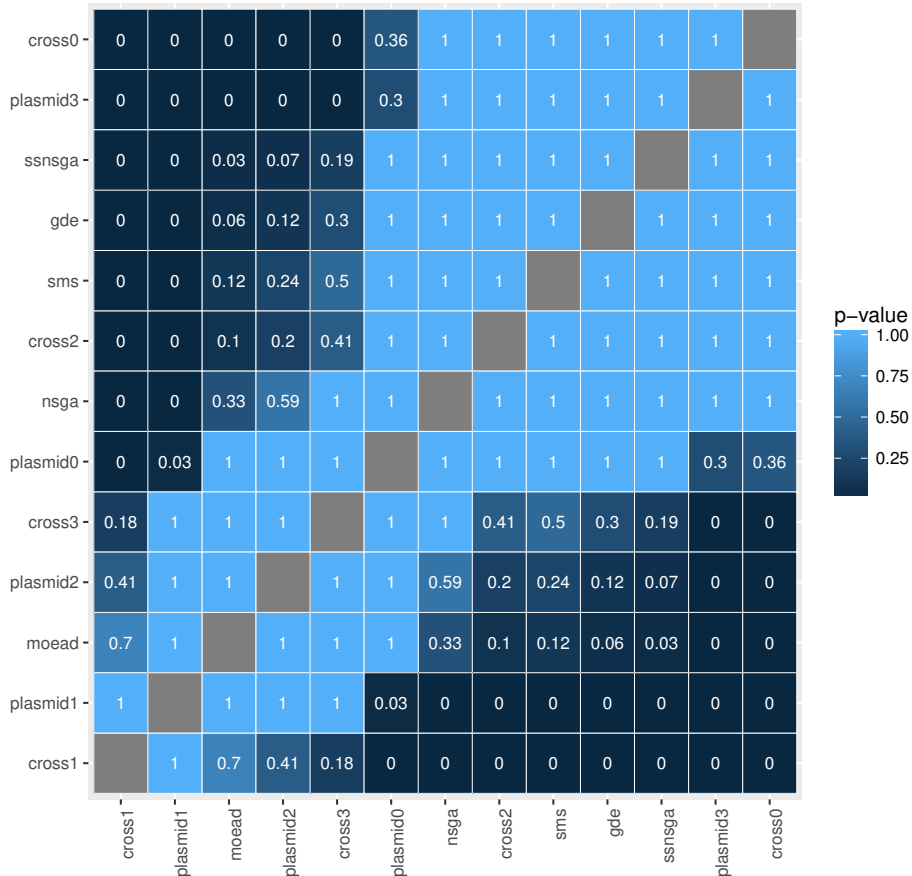
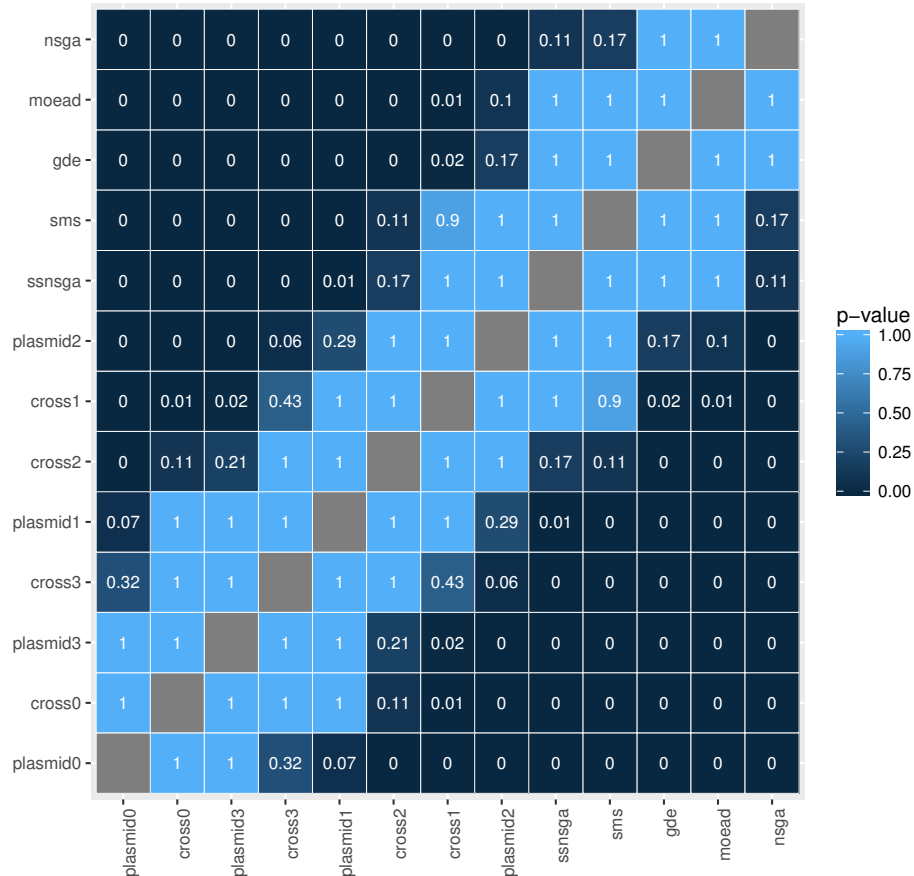


Tabela 81 – Medianas do tempo de processamento no conjunto com $D^k = 50$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	15.439	10.330	12.734	14.291	3.991	3.994	1.994	19.198	13.833	9.485	15.221	6.569	6.698
2	14.124	9.433	11.795	13.530	3.905	3.500	1.744	18.228	13.311	9.135	14.281	6.302	6.381
3	14.414	9.777	11.749	13.712	3.976	3.733	1.879	18.818	13.684	8.957	14.424	6.469	6.548
4	16.091	11.286	12.434	14.140	4.112	3.688	2.059	19.985	15.889	9.086	15.023	6.699	6.768
5	15.040	10.371	12.191	14.318	4.136	3.646	1.848	18.409	13.725	8.954	14.829	6.486	6.579
6	14.949	10.302	11.999	13.871	4.235	3.792	1.848	18.405	13.794	9.240	14.727	6.504	6.582
7	14.755	9.912	12.150	13.937	3.985	3.703	1.860	18.825	13.698	8.983	15.246	6.455	6.521
8	16.469	10.323	13.362	14.723	4.088	4.218	2.136	20.719	14.494	10.447	15.520	6.808	6.841
9	15.942	10.491	13.113	14.513	4.191	4.035	2.008	19.336	14.043	9.900	15.643	6.736	6.822
10	15.093	9.713	12.685	13.935	4.060	3.996	1.967	19.356	13.675	9.638	15.301	6.605	6.630
11	15.881	10.320	13.277	14.449	4.258	4.280	2.108	19.816	14.114	9.977	15.447	6.771	6.814
12	16.019	10.050	13.175	14.679	4.180	4.026	2.069	20.477	14.178	9.969	15.704	6.712	6.783
13	15.132	10.259	12.344	14.072	4.227	3.824	1.881	18.759	13.553	9.581	14.794	6.580	6.632
14	15.742	10.470	13.116	14.170	4.177	4.150	2.017	19.253	13.929	9.737	15.303	6.651	6.751
15	15.074	10.220	12.113	13.796	4.116	3.824	1.895	18.291	13.661	9.199	14.915	6.486	6.587
16	15.349	10.196	12.485	14.411	4.116	3.888	1.951	19.351	13.980	9.547	15.568	6.605	6.679
17	15.473	10.043	13.097	14.697	4.063	4.097	2.026	19.364	13.793	9.784	14.903	6.701	6.753
18	14.963	9.791	12.622	14.302	3.908	3.918	1.978	19.449	13.673	9.338	15.286	6.590	6.607
19	15.344	10.039	12.762	14.545	3.834	4.036	2.008	19.707	13.990	10.028	16.095	6.632	6.708

Figura 73 – *Posthoc* de Shaffer para o Tempo no conjunto com $D^k = 50$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 82 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $W = 10$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.083	0.135	0.308	0.256	0.125	0.231	0.167	0.128	0.128	0.308	0.256	0.167	0.150
2	0.077	0.132	0.077	0.115	0.123	0.171	0.154	0.083	0.139	0.115	0.115	0.154	0.146
3	0.136	0.136	0.273	0.273	0.136	0.182	0.172	0.136	0.136	0.303	0.303	0.136	0.152
4	0.056	0.103	0.091	0.113	0.105	0.194	0.167	0.111	0.111	0.167	0.097	0.167	0.156
5	0.095	0.152	0.300	0.257	0.143	0.233	0.171	0.143	0.143	0.314	0.271	0.190	0.143
6	0.083	0.112	0.222	0.222	0.148	0.208	0.167	0.111	0.118	0.222	0.222	0.167	0.167
7	0.048	0.103	0.095	0.112	0.111	0.148	0.148	0.111	0.143	0.143	0.111	0.143	0.143
8	0.080	0.138	0.190	0.167	0.112	0.179	0.160	0.120	0.120	0.202	0.167	0.160	0.160
9	0.086	0.090	0.257	0.143	0.130	0.200	0.143	0.114	0.114	0.257	0.229	0.171	0.140
10	0.086	0.086	0.314	0.200	0.114	0.171	0.167	0.114	0.143	0.343	0.229	0.167	0.167
11	0.065	0.095	0.097	0.097	0.104	0.148	0.149	0.097	0.125	0.129	0.097	0.129	0.129
12	0.050	0.100	0.083	0.084	0.086	0.185	0.175	0.111	0.125	0.148	0.083	0.174	0.167
13	0.091	0.119	0.242	0.094	0.121	0.220	0.182	0.136	0.136	0.242	0.121	0.182	0.177
14	0.045	0.101	0.130	0.129	0.105	0.174	0.173	0.130	0.130	0.130	0.130	0.132	0.130
15	0.070	0.145	0.070	0.116	0.109	0.150	0.150	0.113	0.116	0.116	0.100	0.140	0.150
16	0.095	0.178	0.250	0.250	0.107	0.214	0.179	0.143	0.143	0.250	0.232	0.179	0.143
17	0.056	0.091	0.111	0.111	0.111	0.194	0.194	0.105	0.158	0.139	0.111	0.167	0.158
18	0.075	0.140	0.175	0.175	0.125	0.182	0.150	0.125	0.136	0.175	0.175	0.150	0.150
19	0.080	0.091	0.160	0.160	0.120	0.200	0.160	0.120	0.136	0.160	0.160	0.160	0.160
20	0.095	0.126	0.133	0.103	0.133	0.238	0.190	0.140	0.143	0.167	0.133	0.190	0.190

Figura 74 – *Posthoc* de Shaffer para $I_{\epsilon+}$ no conjunto com $W = 10$.

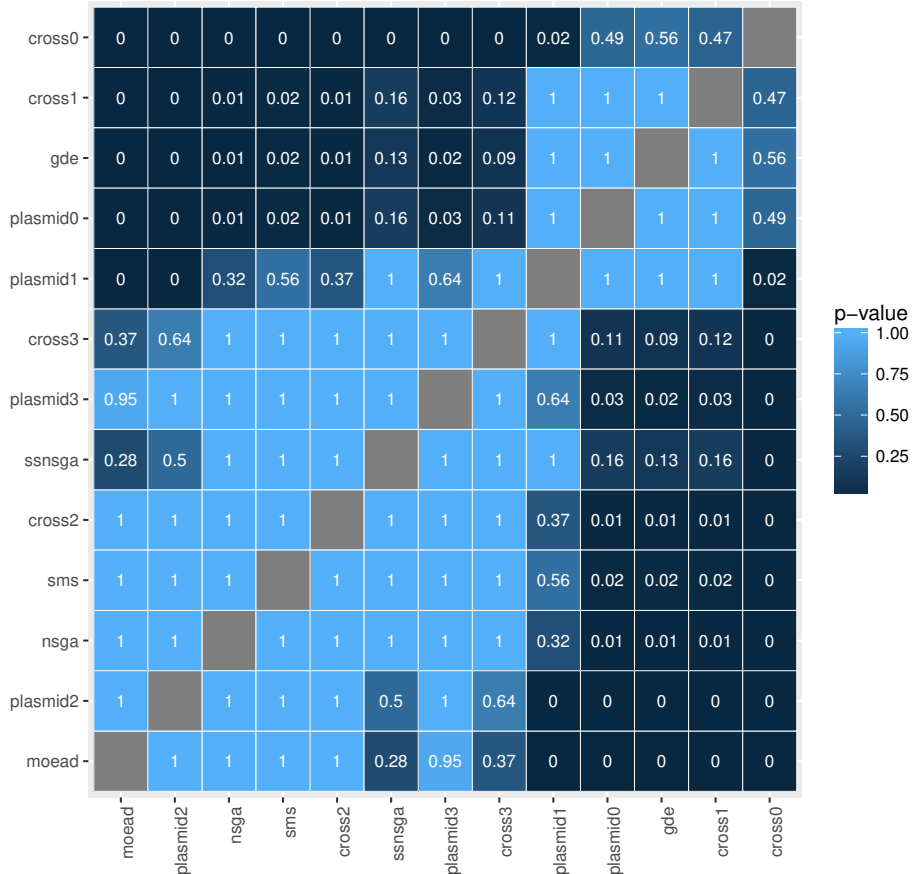
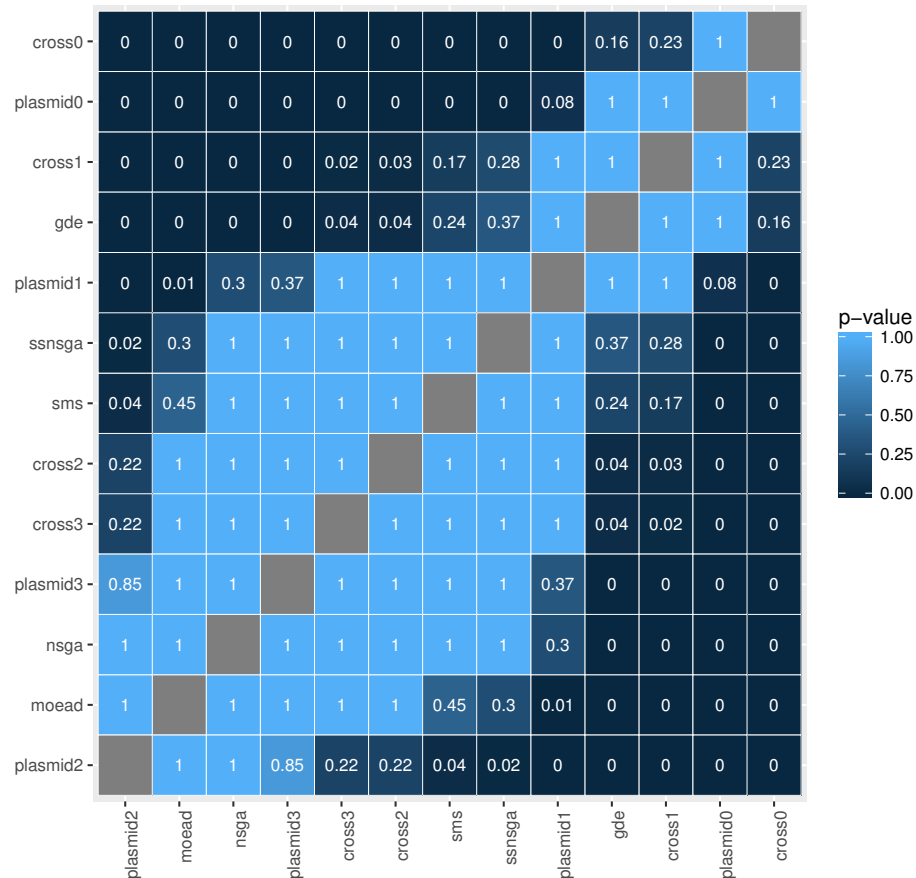


Tabela 83 – Medianas do indicador de qualidade I_H no conjunto com $W = 10$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.120	0.202	0.372	0.322	0.220	0.287	0.258	0.171	0.205	0.388	0.332	0.244	0.238
2	0.140	0.261	0.159	0.248	0.247	0.280	0.286	0.147	0.294	0.223	0.228	0.257	0.252
3	0.076	0.147	0.237	0.253	0.168	0.258	0.223	0.131	0.171	0.295	0.267	0.196	0.207
4	0.062	0.156	0.136	0.164	0.139	0.233	0.225	0.113	0.169	0.274	0.157	0.199	0.196
5	0.126	0.229	0.355	0.333	0.229	0.313	0.251	0.189	0.219	0.407	0.318	0.244	0.232
6	0.096	0.153	0.266	0.289	0.180	0.225	0.199	0.105	0.163	0.293	0.273	0.192	0.185
7	0.114	0.175	0.159	0.217	0.201	0.261	0.267	0.151	0.230	0.193	0.233	0.241	0.227
8	0.130	0.215	0.255	0.254	0.205	0.281	0.255	0.179	0.210	0.294	0.259	0.251	0.251
9	0.060	0.159	0.334	0.222	0.194	0.245	0.220	0.136	0.160	0.373	0.324	0.203	0.198
10	0.087	0.145	0.341	0.225	0.170	0.234	0.234	0.136	0.223	0.368	0.248	0.210	0.206
11	0.091	0.145	0.122	0.160	0.177	0.235	0.219	0.131	0.182	0.192	0.161	0.191	0.196
12	0.083	0.149	0.179	0.165	0.162	0.234	0.239	0.131	0.185	0.297	0.203	0.222	0.214
13	0.092	0.174	0.325	0.180	0.197	0.274	0.259	0.158	0.192	0.366	0.208	0.241	0.239
14	0.081	0.166	0.224	0.233	0.146	0.220	0.215	0.138	0.202	0.281	0.214	0.201	0.193
15	0.095	0.232	0.158	0.190	0.170	0.256	0.254	0.169	0.203	0.257	0.185	0.233	0.241
16	0.099	0.230	0.292	0.330	0.182	0.288	0.262	0.203	0.253	0.362	0.318	0.238	0.240
17	0.133	0.183	0.185	0.185	0.197	0.285	0.272	0.185	0.229	0.241	0.188	0.249	0.255
18	0.092	0.178	0.242	0.270	0.170	0.246	0.218	0.161	0.199	0.277	0.246	0.203	0.206
19	0.087	0.137	0.270	0.277	0.177	0.257	0.228	0.141	0.185	0.306	0.280	0.205	0.211
20	0.111	0.223	0.220	0.210	0.210	0.302	0.272	0.167	0.231	0.305	0.239	0.259	0.252

Figura 75 – *Posthoc* de Shaffer para I_H no conjunto com $W = 10$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 84 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 10$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.077	1.135	1.308	1.256	1.115	1.206	1.148	1.123	1.125	1.308	1.256	1.148	1.129
2	1.077	1.132	1.077	1.115	1.121	1.167	1.154	1.083	1.134	1.115	1.115	1.148	1.136
3	1.091	1.100	1.273	1.273	1.100	1.174	1.146	1.130	1.135	1.303	1.303	1.136	1.136
4	1.056	1.103	1.091	1.113	1.103	1.194	1.164	1.105	1.104	1.167	1.092	1.167	1.148
5	1.087	1.152	1.300	1.257	1.143	1.208	1.143	1.128	1.143	1.314	1.271	1.162	1.143
6	1.080	1.111	1.222	1.222	1.143	1.200	1.160	1.107	1.108	1.222	1.222	1.160	1.160
7	1.048	1.103	1.095	1.112	1.108	1.143	1.143	1.103	1.143	1.138	1.111	1.143	1.136
8	1.077	1.138	1.190	1.167	1.103	1.162	1.153	1.112	1.120	1.202	1.167	1.143	1.152
9	1.083	1.090	1.257	1.143	1.118	1.197	1.143	1.114	1.113	1.257	1.229	1.163	1.135
10	1.086	1.086	1.314	1.200	1.105	1.160	1.158	1.105	1.132	1.343	1.229	1.158	1.149
11	1.065	1.095	1.097	1.097	1.104	1.138	1.142	1.091	1.125	1.129	1.097	1.121	1.123
12	1.047	1.100	1.083	1.084	1.086	1.185	1.167	1.103	1.118	1.148	1.083	1.161	1.160
13	1.083	1.119	1.242	1.094	1.116	1.208	1.167	1.136	1.136	1.242	1.121	1.174	1.160
14	1.045	1.101	1.130	1.125	1.101	1.167	1.158	1.130	1.130	1.130	1.130	1.130	1.130
15	1.068	1.145	1.070	1.116	1.109	1.150	1.150	1.105	1.114	1.116	1.100	1.137	1.143
16	1.087	1.178	1.250	1.250	1.107	1.193	1.150	1.138	1.143	1.250	1.232	1.151	1.143
17	1.056	1.091	1.111	1.111	1.105	1.184	1.179	1.105	1.158	1.139	1.111	1.158	1.158
18	1.073	1.140	1.175	1.175	1.122	1.171	1.146	1.122	1.136	1.175	1.175	1.146	1.146
19	1.077	1.089	1.160	1.160	1.115	1.200	1.160	1.120	1.136	1.160	1.160	1.160	1.160
20	1.087	1.126	1.133	1.103	1.133	1.219	1.182	1.130	1.143	1.167	1.133	1.182	1.182

Figura 76 – *Posthoc* de Shaffer para I_ϵ no conjunto com $W = 10$.

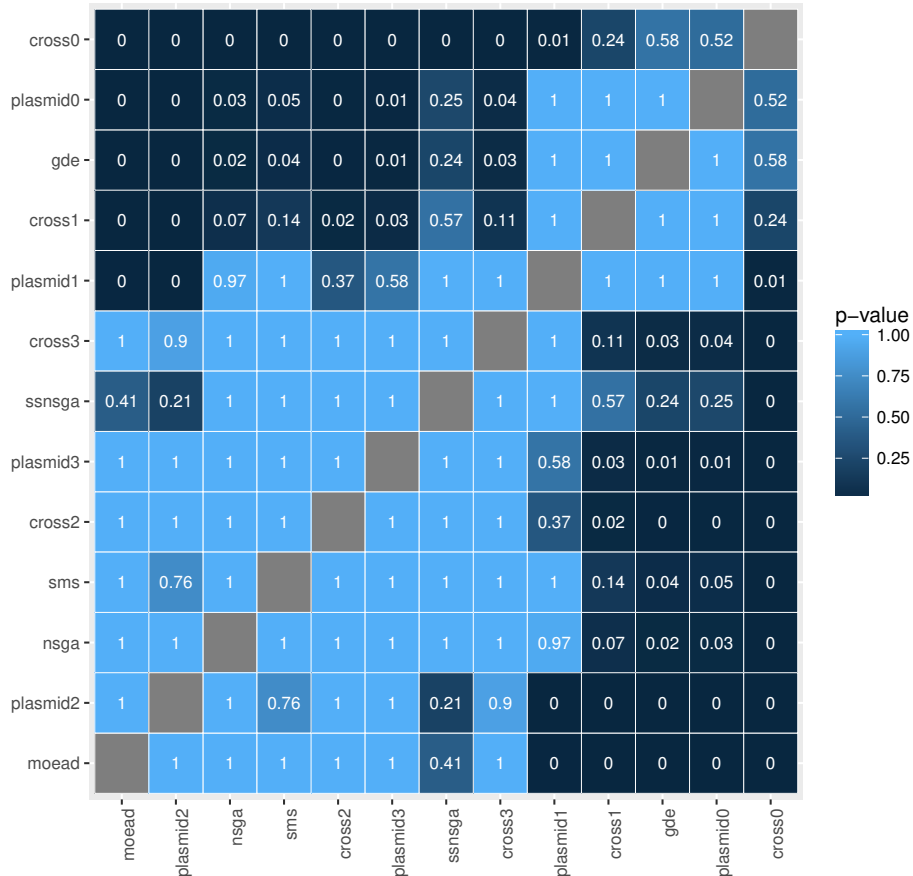
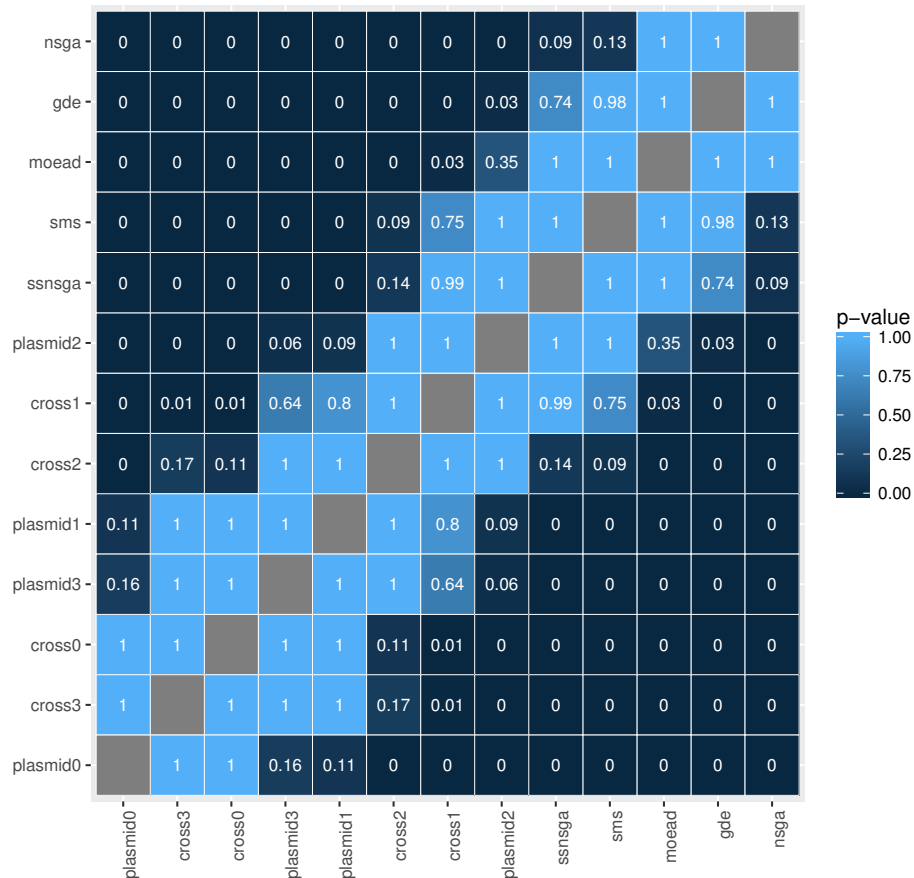


Tabela 85 – Medianas do tempo de processamento no conjunto com $W = 10$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.941	1.409	1.559	1.885	0.581	0.730	0.291	2.230	1.870	1.082	1.805	1.031	1.042
2	1.940	1.364	1.630	1.997	0.724	0.733	0.283	2.216	1.794	1.114	1.935	1.036	1.035
3	1.987	1.441	1.600	1.935	0.586	0.719	0.296	2.310	1.976	1.063	1.867	1.038	1.053
4	1.893	1.346	1.626	1.894	0.576	0.703	0.264	2.200	1.790	1.105	1.842	1.014	1.020
5	1.926	1.370	1.607	1.908	0.553	0.693	0.265	2.211	1.846	1.100	1.871	1.012	1.018
6	1.875	1.342	1.547	1.840	0.564	0.683	0.264	2.146	1.794	1.073	1.725	1.003	1.011
7	1.941	1.373	1.664	1.999	0.675	0.731	0.278	2.202	1.832	1.129	1.890	1.027	1.028
8	2.006	1.423	1.696	2.038	0.565	0.747	0.297	2.388	1.948	1.187	1.973	1.043	1.051
9	1.865	1.346	1.545	1.867	0.576	0.708	0.267	2.148	1.759	1.071	1.738	1.015	1.018
10	1.910	1.379	1.575	1.916	0.562	0.707	0.275	2.244	1.819	1.084	1.836	1.027	1.027
11	1.870	1.382	1.542	1.876	0.592	0.693	0.275	2.164	1.854	1.069	1.773	1.010	1.016
12	1.844	1.321	1.559	1.883	0.582	0.694	0.260	2.142	1.744	1.098	1.750	0.999	1.002
13	1.922	1.385	1.616	1.908	0.568	0.710	0.270	2.254	1.826	1.116	1.877	1.023	1.028
14	1.917	1.349	1.641	1.905	0.645	0.703	0.265	2.166	1.776	1.101	1.811	1.011	1.018
15	1.954	1.388	1.630	1.987	0.589	0.714	0.279	2.278	1.854	1.124	1.974	1.027	1.035
16	1.912	1.364	1.599	1.893	0.569	0.695	0.267	2.203	1.810	1.108	1.851	1.005	1.016
17	1.939	1.404	1.612	1.904	0.625	0.720	0.281	2.203	1.856	1.107	1.812	1.025	1.026
18	1.885	1.322	1.590	1.843	0.566	0.656	0.248	2.186	1.787	1.085	1.767	0.994	0.999
19	1.919	1.384	1.644	1.920	0.608	0.738	0.287	2.212	1.857	1.127	1.831	1.028	1.034
20	1.949	1.385	1.657	2.003	0.571	0.715	0.280	2.256	1.836	1.142	1.904	1.034	1.042

Figura 77 – *Posthoc* de Shaffer para o Tempo no conjunto com $W = 10$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 86 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $W = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.083	0.161	0.207	0.172	0.138	0.190	0.155	0.138	0.125	0.224	0.181	0.138	0.138
2	0.087	0.140	0.241	0.148	0.111	0.259	0.174	0.148	0.167	0.259	0.204	0.174	0.174
3	0.072	0.141	0.091	0.131	0.135	0.196	0.182	0.136	0.136	0.136	0.136	0.174	0.174
4	0.077	0.154	0.196	0.176	0.129	0.192	0.154	0.121	0.154	0.216	0.196	0.137	0.154
5	0.064	0.097	0.149	0.106	0.106	0.170	0.149	0.106	0.112	0.170	0.128	0.130	0.140
6	0.071	0.153	0.140	0.136	0.149	0.160	0.140	0.120	0.121	0.168	0.140	0.127	0.120
7	0.091	0.196	0.196	0.147	0.142	0.214	0.182	0.143	0.136	0.196	0.143	0.147	0.180
8	0.074	0.158	0.191	0.170	0.136	0.185	0.148	0.128	0.148	0.234	0.191	0.140	0.128
9	0.085	0.166	0.100	0.155	0.140	0.182	0.160	0.136	0.136	0.140	0.136	0.160	0.151
10	0.073	0.160	0.255	0.182	0.127	0.182	0.130	0.125	0.125	0.255	0.209	0.127	0.125
11	0.077	0.169	0.176	0.138	0.120	0.192	0.154	0.137	0.154	0.176	0.127	0.154	0.154
12	0.084	0.155	0.125	0.138	0.141	0.250	0.200	0.150	0.150	0.167	0.104	0.200	0.200
13	0.075	0.165	0.113	0.125	0.115	0.189	0.167	0.132	0.125	0.167	0.125	0.170	0.167
14	0.075	0.166	0.208	0.146	0.122	0.227	0.182	0.146	0.136	0.229	0.171	0.174	0.146
15	0.098	0.176	0.146	0.136	0.150	0.220	0.171	0.150	0.150	0.150	0.146	0.168	0.150
16	0.088	0.160	0.088	0.143	0.123	0.217	0.175	0.158	0.130	0.157	0.130	0.193	0.175
17	0.087	0.192	0.193	0.137	0.136	0.182	0.175	0.136	0.136	0.179	0.140	0.158	0.140
18	0.080	0.139	0.111	0.118	0.119	0.200	0.167	0.130	0.160	0.148	0.093	0.160	0.160
19	0.189	0.151	0.091	0.095	0.189	0.304	0.261	0.189	0.189	0.131	0.130	0.261	0.253
20	0.068	0.143	0.089	0.118	0.123	0.178	0.174	0.130	0.130	0.133	0.089	0.174	0.156

Figura 78 – *Posthoc* de Shaffer para I_{ϵ_+} no conjunto com $W = 20$.

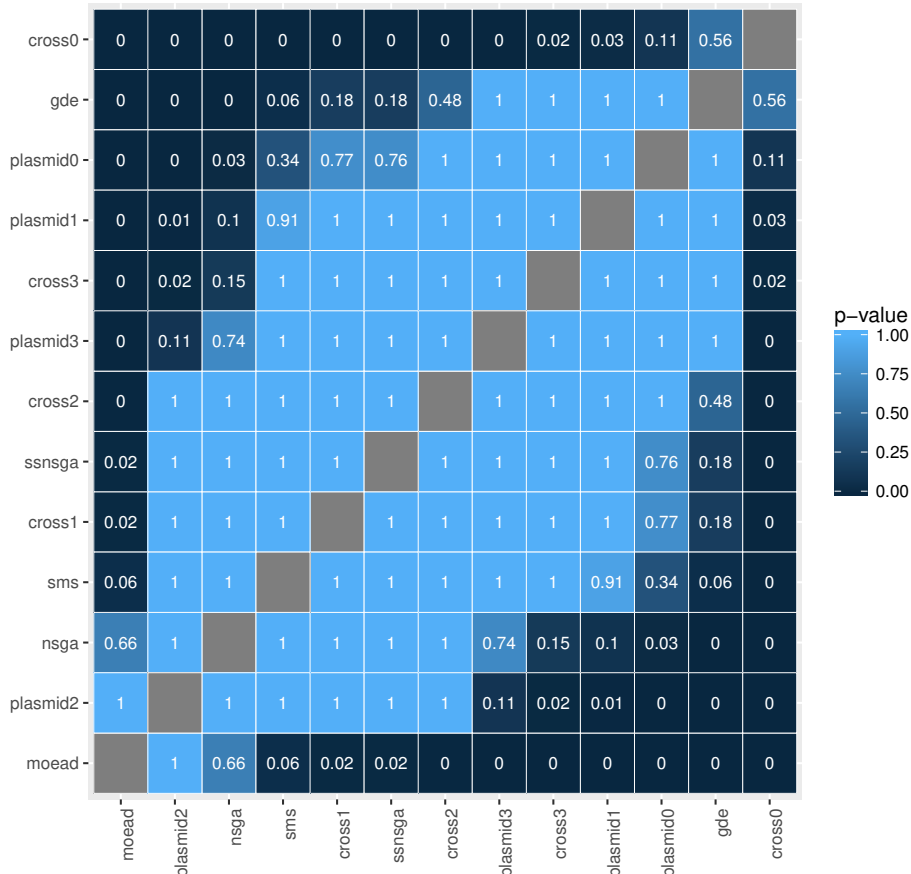
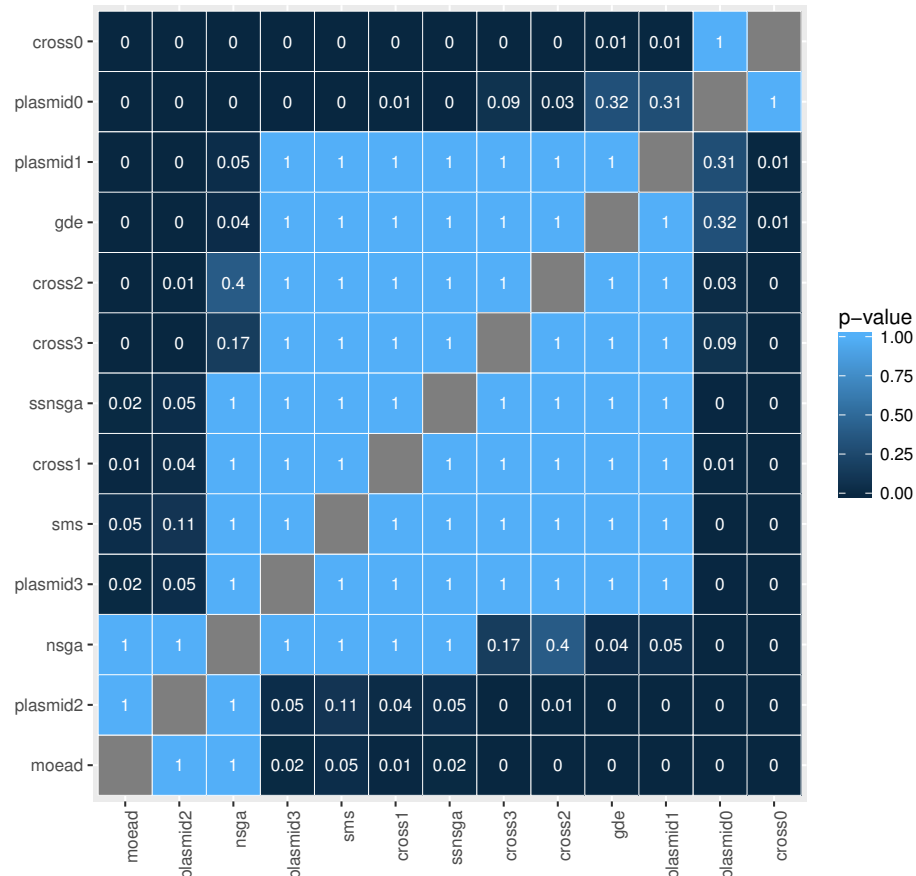


Tabela 87 – Medianas do indicador de qualidade I_H no conjunto com $W = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.121	0.218	0.233	0.239	0.228	0.268	0.236	0.182	0.179	0.297	0.254	0.215	0.217
2	0.120	0.224	0.282	0.208	0.190	0.287	0.237	0.180	0.198	0.319	0.282	0.233	0.218
3	0.145	0.213	0.179	0.225	0.223	0.310	0.309	0.187	0.245	0.252	0.218	0.274	0.275
4	0.147	0.234	0.265	0.274	0.252	0.283	0.251	0.192	0.215	0.333	0.290	0.230	0.229
5	0.126	0.161	0.221	0.187	0.185	0.248	0.249	0.170	0.168	0.266	0.210	0.217	0.213
6	0.131	0.239	0.235	0.245	0.261	0.250	0.240	0.177	0.205	0.301	0.240	0.214	0.202
7	0.154	0.286	0.243	0.242	0.257	0.321	0.279	0.200	0.231	0.304	0.249	0.255	0.255
8	0.127	0.251	0.268	0.276	0.238	0.260	0.235	0.165	0.229	0.334	0.277	0.222	0.216
9	0.128	0.254	0.209	0.255	0.222	0.274	0.260	0.180	0.219	0.275	0.240	0.237	0.244
10	0.128	0.240	0.316	0.266	0.239	0.278	0.230	0.165	0.205	0.343	0.279	0.216	0.217
11	0.137	0.249	0.263	0.248	0.227	0.282	0.254	0.180	0.227	0.291	0.262	0.241	0.240
12	0.160	0.241	0.192	0.177	0.246	0.331	0.322	0.209	0.252	0.286	0.199	0.293	0.293
13	0.116	0.209	0.202	0.227	0.188	0.271	0.243	0.176	0.217	0.298	0.230	0.234	0.233
14	0.124	0.239	0.279	0.254	0.196	0.287	0.258	0.177	0.193	0.347	0.285	0.241	0.227
15	0.147	0.269	0.252	0.277	0.218	0.317	0.289	0.187	0.252	0.304	0.280	0.260	0.262
16	0.167	0.271	0.167	0.180	0.242	0.338	0.307	0.222	0.273	0.243	0.205	0.292	0.287
17	0.134	0.283	0.277	0.238	0.235	0.277	0.256	0.174	0.225	0.321	0.248	0.236	0.236
18	0.123	0.229	0.177	0.185	0.201	0.262	0.253	0.168	0.236	0.246	0.188	0.234	0.233
19	0.289	0.318	0.161	0.159	0.344	0.477	0.425	0.306	0.366	0.246	0.193	0.414	0.398
20	0.165	0.258	0.216	0.191	0.228	0.291	0.293	0.200	0.268	0.277	0.193	0.276	0.268

Figura 79 – *Posthoc* de Shaffer para I_H no conjunto com $W = 20$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 88 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.067	1.161	1.207	1.172	1.134	1.171	1.136	1.126	1.125	1.224	1.181	1.125	1.125
2	1.081	1.140	1.241	1.148	1.109	1.226	1.174	1.132	1.146	1.259	1.204	1.174	1.161
3	1.072	1.141	1.091	1.131	1.135	1.194	1.182	1.136	1.136	1.136	1.136	1.170	1.170
4	1.077	1.154	1.196	1.176	1.124	1.172	1.148	1.115	1.145	1.216	1.196	1.125	1.136
5	1.064	1.097	1.149	1.106	1.104	1.167	1.143	1.103	1.105	1.157	1.128	1.130	1.133
6	1.071	1.153	1.140	1.136	1.149	1.153	1.133	1.120	1.120	1.168	1.140	1.120	1.120
7	1.089	1.196	1.196	1.147	1.139	1.192	1.168	1.136	1.136	1.196	1.143	1.137	1.168
8	1.074	1.158	1.191	1.170	1.136	1.160	1.132	1.118	1.148	1.234	1.191	1.124	1.121
9	1.085	1.166	1.100	1.155	1.140	1.176	1.157	1.136	1.136	1.140	1.127	1.157	1.140
10	1.069	1.160	1.255	1.182	1.125	1.164	1.125	1.125	1.125	1.255	1.209	1.125	1.125
11	1.077	1.169	1.176	1.138	1.120	1.178	1.154	1.127	1.143	1.176	1.127	1.148	1.154
12	1.084	1.155	1.125	1.138	1.140	1.240	1.200	1.150	1.150	1.167	1.104	1.200	1.195
13	1.074	1.165	1.113	1.125	1.114	1.185	1.160	1.130	1.125	1.167	1.120	1.167	1.160
14	1.075	1.166	1.208	1.146	1.121	1.216	1.170	1.137	1.136	1.229	1.167	1.163	1.137
15	1.093	1.176	1.146	1.136	1.149	1.214	1.158	1.150	1.150	1.150	1.140	1.152	1.150
16	1.088	1.160	1.088	1.143	1.121	1.209	1.174	1.155	1.130	1.147	1.130	1.187	1.174
17	1.081	1.192	1.193	1.137	1.136	1.180	1.167	1.136	1.136	1.175	1.140	1.145	1.136
18	1.080	1.139	1.111	1.118	1.116	1.193	1.160	1.123	1.160	1.148	1.093	1.160	1.159
19	1.189	1.151	1.090	1.095	1.189	1.304	1.259	1.187	1.189	1.130	1.128	1.261	1.241
20	1.068	1.143	1.089	1.118	1.123	1.178	1.174	1.130	1.130	1.133	1.089	1.167	1.151

Figura 80 – *Posthoc* de Shaffer para I_ϵ no conjunto com $W = 20$.

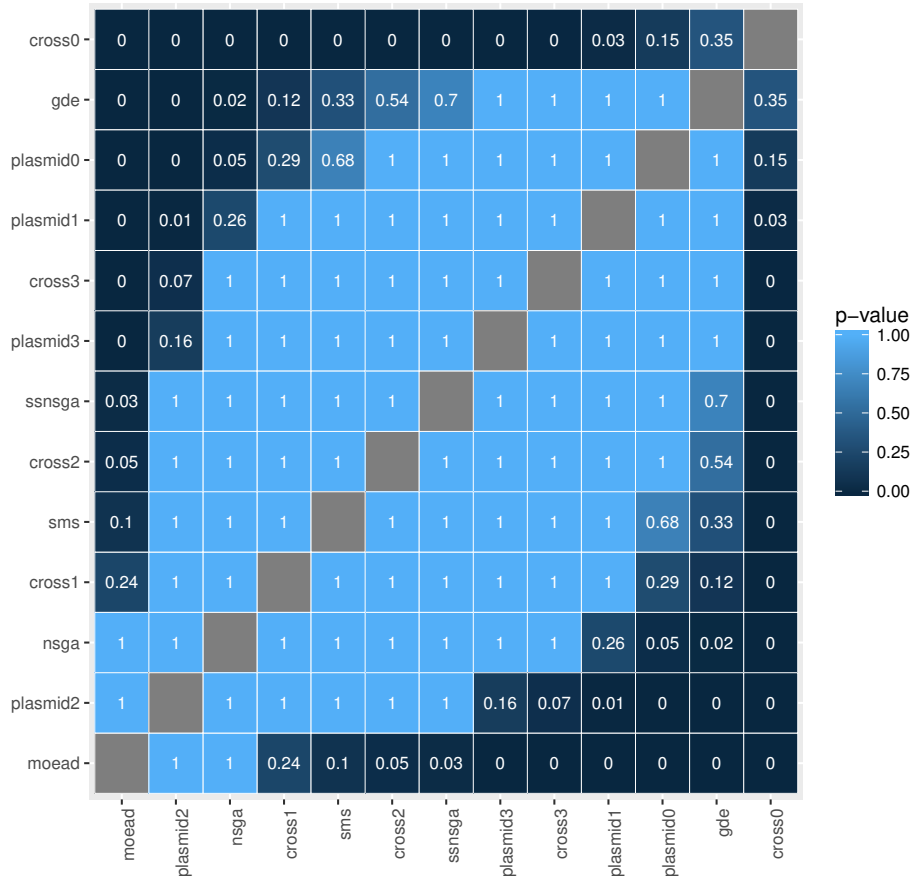
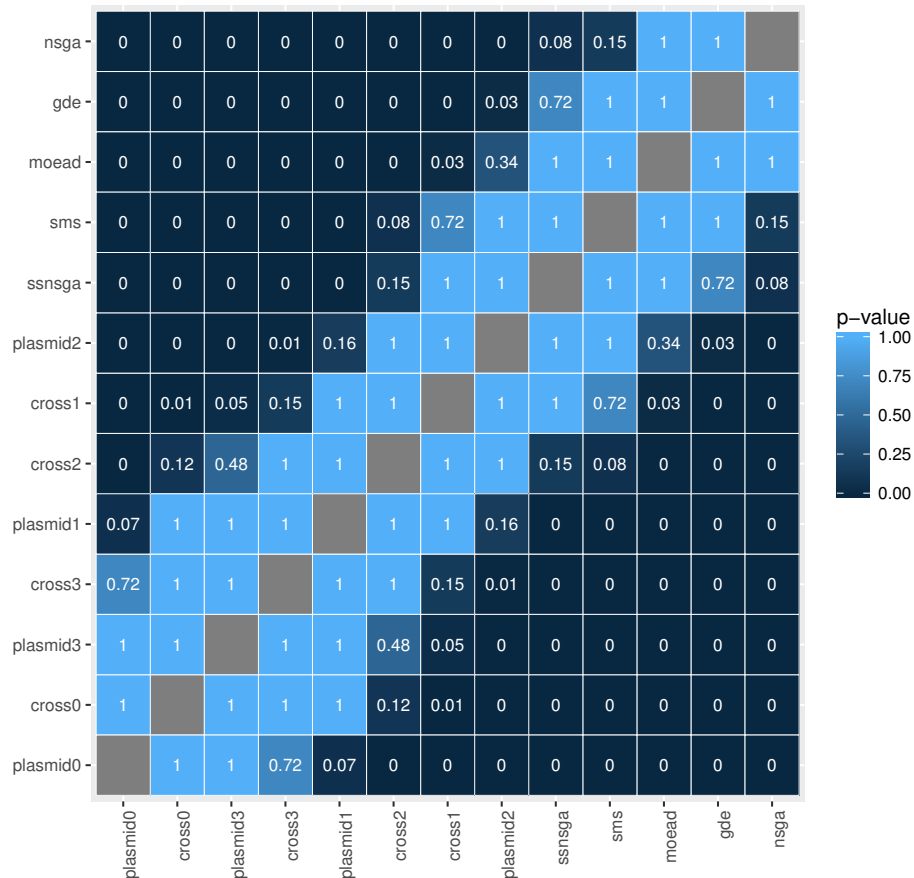


Tabela 89 – Medianas do tempo de processamento no conjunto com $W = 20$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	3.709	2.741	3.038	3.557	1.095	1.362	0.550	4.471	3.634	2.123	3.477	1.881	1.902
2	3.690	2.679	3.024	3.610	1.100	1.365	0.549	4.376	3.625	2.139	3.885	1.865	1.891
3	3.750	2.668	3.182	3.654	1.114	1.436	0.565	4.660	3.613	2.312	4.026	1.882	1.893
4	3.656	2.673	3.032	3.491	1.125	1.345	0.536	4.276	3.493	2.172	3.589	1.869	1.882
5	3.566	2.614	2.943	3.513	1.063	1.347	0.511	4.105	3.415	2.071	3.450	1.814	1.822
6	3.722	2.712	3.049	3.567	1.103	1.387	0.549	4.323	3.553	2.106	3.640	1.878	1.899
7	3.784	2.770	3.134	3.598	1.114	1.411	0.569	4.509	3.649	2.207	3.608	1.910	1.936
8	3.791	2.741	3.134	3.701	1.123	1.384	0.549	4.493	3.580	2.191	3.742	1.879	1.903
9	3.669	2.644	3.046	3.509	1.062	1.370	0.541	4.361	3.485	2.155	3.482	1.844	1.867
10	3.873	2.813	3.307	3.793	1.100	1.477	0.610	4.717	3.744	2.231	3.930	1.928	1.955
11	3.781	2.771	3.115	3.657	1.123	1.426	0.565	4.523	3.685	2.218	3.666	1.887	1.917
12	3.626	2.602	3.056	3.667	1.039	1.385	0.539	4.417	3.451	2.139	3.677	1.826	1.848
13	3.818	2.703	3.190	3.717	1.102	1.455	0.567	4.584	3.621	2.234	3.864	1.883	1.898
14	3.717	2.665	3.130	3.657	1.093	1.448	0.561	4.528	3.583	2.220	3.916	1.872	1.882
15	3.651	2.623	3.071	3.656	1.070	1.366	0.524	4.328	3.464	2.155	3.525	1.829	1.846
16	3.641	2.629	3.112	3.703	1.096	1.363	0.524	4.552	3.493	2.226	3.782	1.837	1.848
17	3.635	2.652	3.009	3.533	1.086	1.328	0.528	4.345	3.499	2.091	3.580	1.846	1.870
18	3.724	2.636	3.091	3.678	1.113	1.357	0.526	4.331	3.484	2.115	3.598	1.850	1.871
19	3.560	2.605	3.007	3.563	1.193	1.368	0.530	4.343	3.470	2.080	3.581	1.828	1.836
20	3.723	2.640	3.155	3.676	1.164	1.379	0.523	4.310	3.517	2.189	3.568	1.840	1.860

Figura 81 – *Posthoc* de Shaffer para o Tempo no conjunto com $W = 20$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 90 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $W = 30$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.080	0.176	0.207	0.138	0.146	0.172	0.135	0.115	0.115	0.207	0.172	0.115	0.115
2	0.088	0.183	0.175	0.158	0.160	0.246	0.185	0.148	0.148	0.175	0.175	0.158	0.167
3	0.181	0.162	0.147	0.140	0.206	0.296	0.259	0.235	0.222	0.167	0.139	0.259	0.255
4	0.096	0.202	0.172	0.139	0.165	0.207	0.172	0.148	0.148	0.207	0.190	0.155	0.148
5	0.097	0.186	0.246	0.175	0.152	0.211	0.175	0.154	0.161	0.289	0.211	0.172	0.166
6	0.095	0.196	0.207	0.190	0.138	0.241	0.197	0.173	0.155	0.310	0.207	0.188	0.173
7	0.128	0.152	0.085	0.135	0.148	0.255	0.191	0.185	0.148	0.149	0.111	0.195	0.195
8	0.088	0.178	0.228	0.123	0.134	0.211	0.193	0.193	0.160	0.246	0.175	0.171	0.175
9	0.174	0.189	0.087	0.129	0.179	0.304	0.245	0.217	0.214	0.143	0.130	0.251	0.225
10	0.091	0.168	0.233	0.190	0.155	0.198	0.177	0.155	0.137	0.259	0.233	0.172	0.172
11	0.087	0.182	0.167	0.130	0.143	0.222	0.174	0.174	0.156	0.204	0.167	0.174	0.167
12	0.094	0.203	0.200	0.149	0.156	0.218	0.182	0.163	0.148	0.236	0.200	0.173	0.151
13	0.094	0.186	0.189	0.170	0.189	0.245	0.200	0.189	0.200	0.264	0.208	0.200	0.200
14	0.109	0.196	0.164	0.129	0.154	0.255	0.183	0.178	0.191	0.182	0.159	0.182	0.182
15	0.083	0.151	0.098	0.132	0.136	0.196	0.176	0.167	0.167	0.157	0.125	0.167	0.162
16	0.078	0.130	0.078	0.124	0.122	0.157	0.157	0.137	0.118	0.124	0.111	0.153	0.148
17	0.093	0.166	0.140	0.166	0.149	0.259	0.222	0.186	0.178	0.198	0.163	0.222	0.222
18	0.101	0.212	0.293	0.207	0.162	0.250	0.202	0.181	0.167	0.302	0.259	0.181	0.175
19	0.128	0.130	0.128	0.125	0.128	0.277	0.234	0.202	0.174	0.172	0.130	0.234	0.217
20	0.271	0.271	0.146	0.139	0.271	0.308	0.271	0.271	0.271	0.154	0.150	0.271	0.271

Figura 82 – *Posthoc* de Shaffer para I_{ϵ_+} no conjunto com $W = 30$.

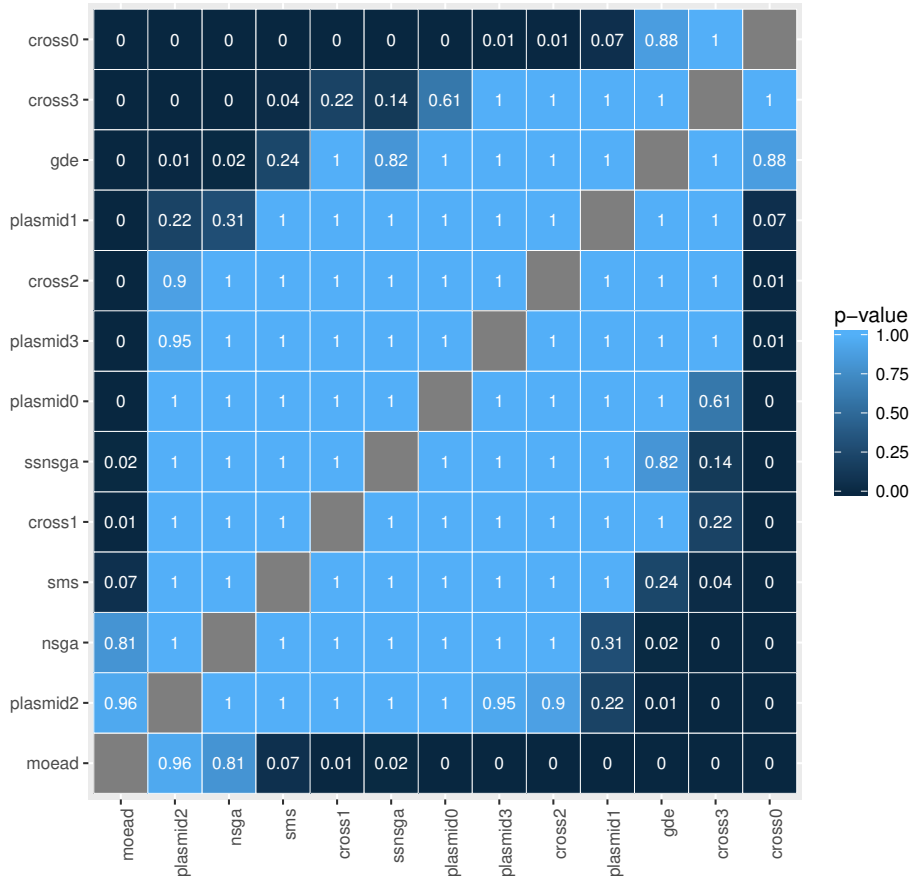
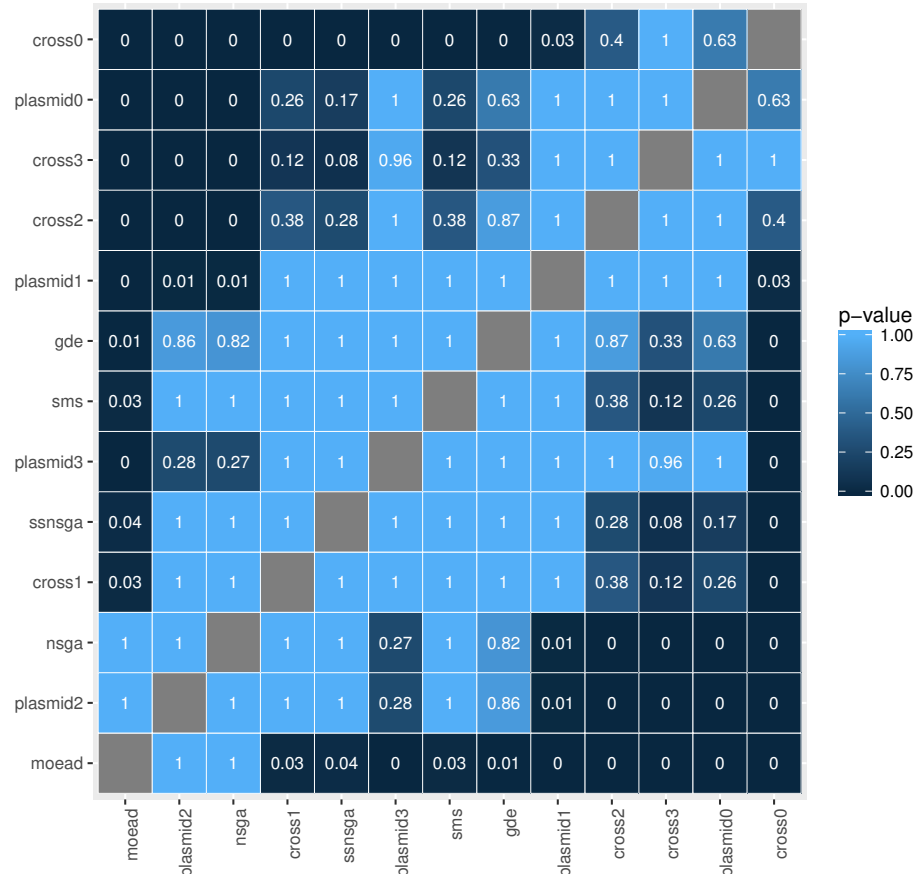


Tabela 91 – Medianas do indicador de qualidade I_H no conjunto com $W = 30$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.151	0.237	0.279	0.266	0.255	0.251	0.231	0.169	0.176	0.304	0.280	0.199	0.202
2	0.158	0.276	0.223	0.259	0.293	0.334	0.282	0.193	0.244	0.292	0.261	0.255	0.260
3	0.276	0.364	0.230	0.206	0.351	0.421	0.403	0.355	0.362	0.289	0.239	0.389	0.390
4	0.176	0.298	0.247	0.247	0.281	0.325	0.289	0.223	0.231	0.338	0.284	0.265	0.271
5	0.157	0.276	0.284	0.247	0.273	0.308	0.264	0.195	0.221	0.365	0.285	0.245	0.250
6	0.152	0.254	0.260	0.281	0.260	0.349	0.294	0.249	0.224	0.363	0.274	0.273	0.274
7	0.259	0.327	0.203	0.243	0.321	0.381	0.369	0.282	0.330	0.301	0.211	0.348	0.357
8	0.143	0.263	0.279	0.212	0.245	0.313	0.281	0.241	0.230	0.363	0.263	0.241	0.243
9	0.227	0.324	0.204	0.175	0.331	0.381	0.348	0.286	0.294	0.268	0.227	0.335	0.324
10	0.155	0.254	0.280	0.252	0.269	0.310	0.275	0.213	0.209	0.378	0.323	0.257	0.253
11	0.141	0.264	0.248	0.227	0.243	0.318	0.257	0.215	0.220	0.318	0.253	0.243	0.249
12	0.155	0.273	0.254	0.256	0.274	0.300	0.269	0.195	0.206	0.330	0.290	0.241	0.237
13	0.170	0.277	0.254	0.284	0.327	0.359	0.319	0.252	0.275	0.376	0.338	0.299	0.300
14	0.178	0.273	0.211	0.211	0.261	0.327	0.300	0.262	0.259	0.302	0.258	0.268	0.266
15	0.143	0.221	0.208	0.209	0.207	0.279	0.267	0.207	0.210	0.279	0.228	0.248	0.244
16	0.169	0.238	0.173	0.166	0.214	0.280	0.289	0.199	0.248	0.226	0.186	0.246	0.246
17	0.190	0.259	0.213	0.246	0.255	0.351	0.336	0.277	0.278	0.328	0.285	0.308	0.304
18	0.149	0.273	0.305	0.281	0.265	0.346	0.305	0.271	0.234	0.413	0.356	0.277	0.279
19	0.231	0.303	0.225	0.221	0.293	0.400	0.385	0.286	0.336	0.299	0.241	0.359	0.356
20	0.343	0.444	0.233	0.169	0.380	0.457	0.452	0.394	0.435	0.274	0.196	0.422	0.431

Figura 83 – *Posthoc* de Shaffer para I_H no conjunto com $W = 30$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 92 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 30$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.080	1.176	1.207	1.138	1.146	1.157	1.117	1.115	1.115	1.207	1.172	1.111	1.115
2	1.086	1.183	1.175	1.158	1.160	1.220	1.179	1.148	1.148	1.175	1.175	1.148	1.149
3	1.176	1.162	1.147	1.140	1.192	1.288	1.250	1.231	1.214	1.159	1.136	1.250	1.250
4	1.096	1.202	1.172	1.139	1.165	1.186	1.155	1.148	1.148	1.207	1.190	1.148	1.148
5	1.091	1.186	1.246	1.175	1.146	1.188	1.161	1.134	1.144	1.289	1.211	1.155	1.155
6	1.090	1.196	1.207	1.190	1.138	1.226	1.175	1.160	1.138	1.310	1.207	1.172	1.166
7	1.128	1.152	1.085	1.135	1.147	1.243	1.191	1.184	1.148	1.149	1.111	1.191	1.189
8	1.080	1.178	1.228	1.123	1.124	1.200	1.169	1.171	1.154	1.246	1.175	1.160	1.160
9	1.172	1.189	1.087	1.129	1.174	1.286	1.239	1.214	1.207	1.143	1.122	1.248	1.217
10	1.087	1.168	1.233	1.190	1.155	1.182	1.158	1.138	1.136	1.259	1.233	1.152	1.149
11	1.085	1.182	1.167	1.130	1.143	1.207	1.167	1.160	1.140	1.204	1.167	1.160	1.150
12	1.092	1.203	1.200	1.149	1.156	1.197	1.163	1.148	1.148	1.236	1.200	1.155	1.148
13	1.090	1.186	1.189	1.170	1.171	1.217	1.195	1.165	1.176	1.264	1.208	1.172	1.184
14	1.098	1.196	1.164	1.129	1.154	1.233	1.174	1.174	1.177	1.182	1.146	1.174	1.174
15	1.083	1.151	1.098	1.132	1.136	1.196	1.169	1.165	1.160	1.157	1.125	1.160	1.158
16	1.078	1.130	1.078	1.124	1.122	1.157	1.157	1.127	1.117	1.115	1.111	1.153	1.143
17	1.093	1.166	1.140	1.166	1.149	1.253	1.214	1.182	1.165	1.195	1.163	1.213	1.214
18	1.097	1.212	1.293	1.207	1.159	1.217	1.176	1.167	1.160	1.302	1.259	1.167	1.167
19	1.128	1.130	1.126	1.125	1.128	1.274	1.234	1.200	1.174	1.168	1.130	1.232	1.215
20	1.271	1.271	1.146	1.139	1.271	1.300	1.271	1.271	1.271	1.154	1.140	1.271	1.271

Figura 84 – *Posthoc* de Shaffer para I_ϵ no conjunto com $W = 30$.

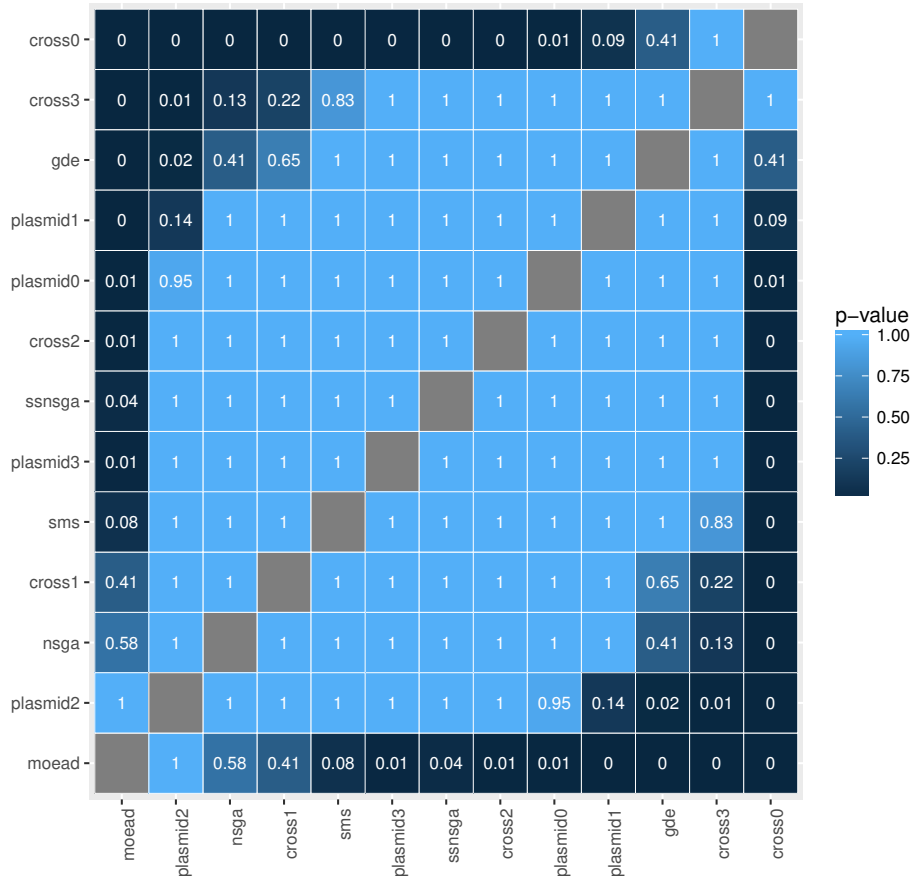
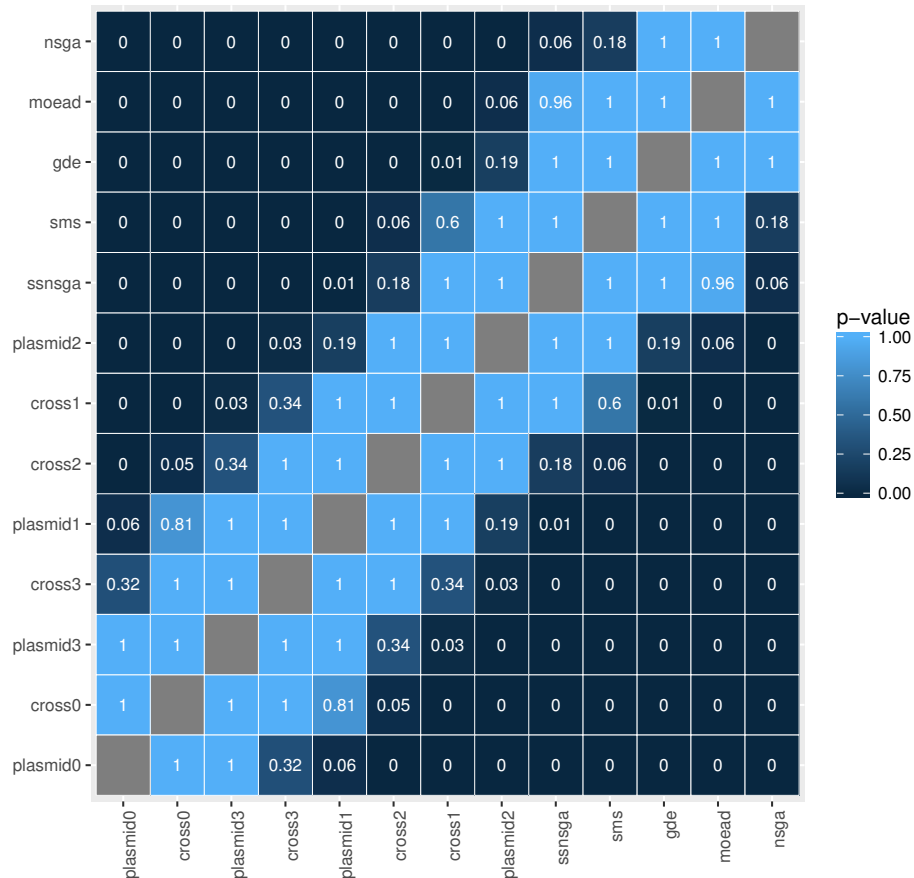


Tabela 93 – Medianas do tempo de processamento no conjunto com $W = 30$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	5.419	3.953	4.463	5.254	1.813	2.061	0.833	6.394	5.246	3.194	5.161	2.776	2.810
2	5.647	4.149	4.574	5.375	2.116	2.064	0.892	6.980	5.342	3.224	5.971	2.866	2.891
3	5.598	4.034	4.558	5.292	1.854	1.991	0.801	6.667	5.405	3.251	5.417	2.772	2.805
4	5.723	4.214	4.603	5.541	2.100	2.047	0.869	6.739	5.406	3.257	5.648	2.889	2.917
5	5.594	4.139	4.453	5.315	2.124	1.881	0.772	6.499	5.290	3.130	5.518	2.818	2.851
6	5.538	4.030	4.506	5.284	2.089	1.970	0.801	6.706	5.324	3.195	5.395	2.760	2.802
7	5.702	4.041	4.627	5.434	2.204	2.042	0.839	6.747	5.248	3.161	5.514	2.798	2.831
8	5.565	4.084	4.545	5.367	2.151	2.010	0.815	6.553	5.347	3.223	5.478	2.788	2.821
9	5.536	4.050	4.480	5.241	2.268	1.932	0.771	6.451	5.209	3.180	5.365	2.752	2.793
10	5.459	4.005	4.451	5.362	2.235	1.934	0.798	6.318	5.191	3.072	5.365	2.767	2.820
11	5.525	4.015	4.536	5.266	2.069	1.966	0.795	6.480	5.275	3.173	5.324	2.744	2.783
12	5.610	4.116	4.549	5.331	1.985	2.027	0.854	6.713	5.374	3.251	5.799	2.845	2.865
13	5.560	4.050	4.533	5.348	2.124	1.899	0.791	6.622	5.235	3.125	5.410	2.806	2.842
14	5.586	3.992	4.608	5.327	2.067	2.039	0.803	6.478	5.325	3.225	5.334	2.764	2.787
15	5.478	3.944	4.534	5.277	1.997	1.995	0.777	6.741	5.128	3.281	5.398	2.747	2.781
16	5.384	3.857	4.465	5.236	1.898	1.931	0.752	6.567	5.005	3.100	5.360	2.719	2.741
17	5.592	3.932	4.715	5.302	2.173	2.024	0.804	6.709	5.243	3.333	5.321	2.760	2.780
18	5.546	4.108	4.506	5.349	2.110	1.996	0.830	6.535	5.263	3.147	5.573	2.809	2.834
19	5.511	3.967	4.575	5.483	2.032	2.030	0.815	6.956	5.142	3.194	5.442	2.751	2.789
20	5.696	4.126	4.688	5.568	2.113	2.026	0.836	6.988	5.287	3.290	5.935	2.814	2.837

Figura 85 – *Posthoc* de Shaffer para o Tempo no conjunto com $W = 30$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 94 – Medianas do indicador de qualidade $I_{\epsilon+}$ no conjunto com $W = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.140	0.195	0.260	0.161	0.161	0.300	0.250	0.218	0.208	0.270	0.200	0.250	0.220
2	0.135	0.162	0.404	0.269	0.223	0.286	0.214	0.214	0.212	0.500	0.423	0.214	0.194
3	0.115	0.178	0.245	0.139	0.196	0.255	0.192	0.192	0.157	0.314	0.216	0.192	0.176
4	0.130	0.218	0.264	0.189	0.208	0.264	0.217	0.217	0.217	0.283	0.264	0.217	0.217
5	0.098	0.210	0.294	0.235	0.235	0.255	0.200	0.200	0.160	0.353	0.373	0.198	0.200
6	0.120	0.200	0.340	0.220	0.220	0.280	0.200	0.200	0.180	0.360	0.290	0.180	0.173
7	0.106	0.173	0.385	0.269	0.200	0.250	0.173	0.196	0.160	0.385	0.308	0.173	0.169
8	0.270	0.270	0.162	0.129	0.270	0.333	0.270	0.270	0.270	0.270	0.185	0.296	0.296
9	0.143	0.164	0.268	0.122	0.206	0.321	0.250	0.217	0.214	0.317	0.195	0.250	0.231
10	0.102	0.196	0.245	0.184	0.164	0.265	0.200	0.200	0.167	0.306	0.265	0.201	0.200
11	0.148	0.200	0.269	0.155	0.231	0.286	0.214	0.214	0.210	0.308	0.250	0.214	0.214
12	0.500	0.861	0.190	0.135	0.524	0.583	0.488	0.548	0.861	0.214	0.190	0.536	0.452
13	0.140	0.219	0.280	0.150	0.180	0.280	0.240	0.200	0.200	0.280	0.220	0.200	0.200
14	0.160	0.204	0.354	0.271	0.229	0.333	0.240	0.245	0.240	0.417	0.375	0.240	0.229
15	0.154	0.143	0.400	0.178	0.178	0.346	0.269	0.231	0.231	0.411	0.244	0.269	0.267
16	0.102	0.174	0.245	0.204	0.184	0.245	0.200	0.184	0.160	0.306	0.327	0.192	0.163
17	0.111	0.148	0.604	0.333	0.271	0.296	0.208	0.222	0.222	0.656	0.438	0.211	0.191
18	0.125	0.179	0.333	0.167	0.188	0.280	0.200	0.208	0.200	0.375	0.250	0.219	0.200
19	0.106	0.175	0.277	0.213	0.191	0.250	0.208	0.191	0.167	0.319	0.255	0.200	0.191
20	0.095	0.203	0.318	0.236	0.167	0.236	0.182	0.164	0.143	0.327	0.309	0.179	0.168

Figura 86 – *Posthoc* de Shaffer para $I_{\epsilon+}$ no conjunto com $W = 40$.

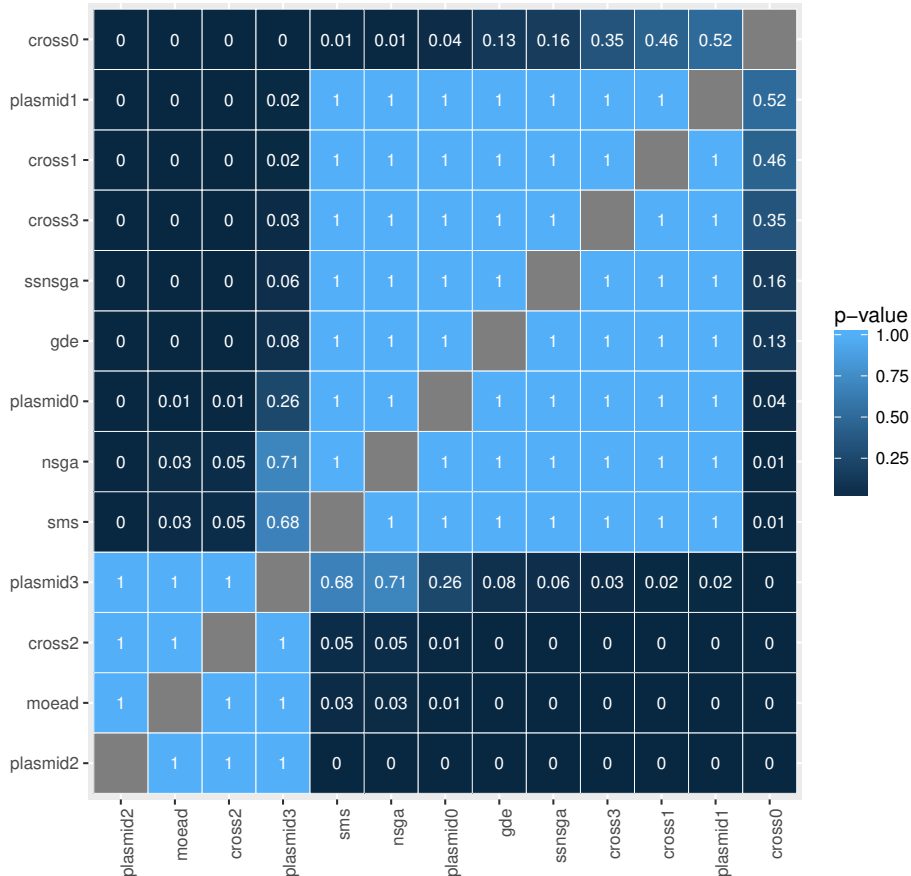
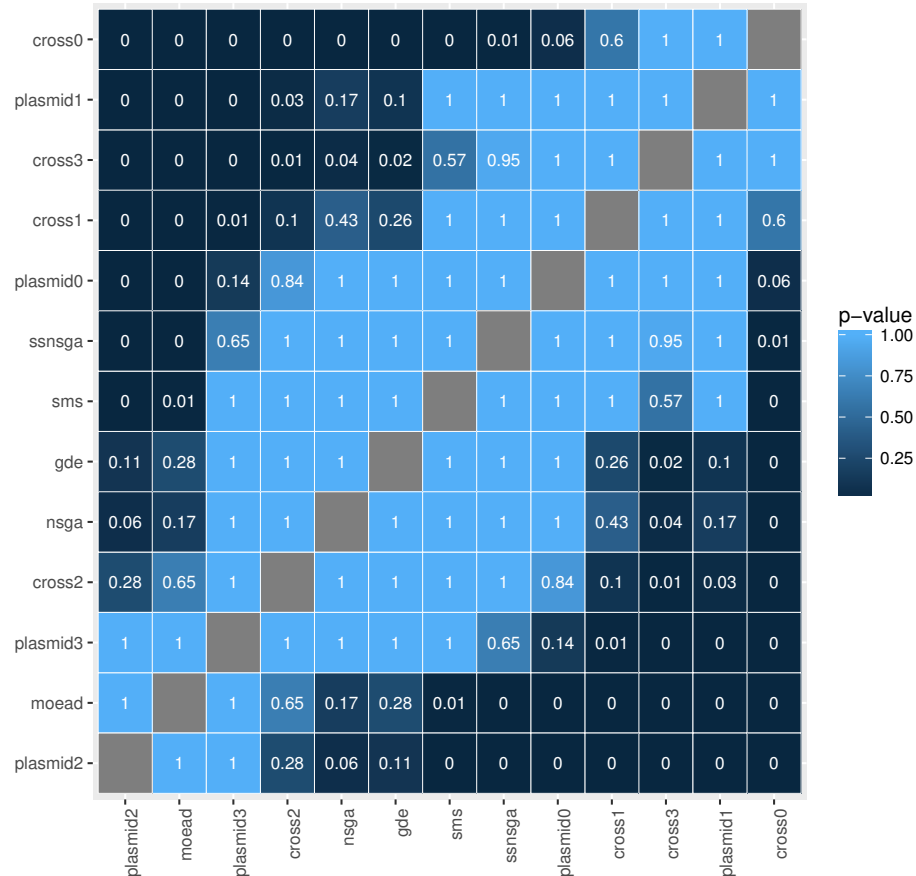


Tabela 95 – Medianas do indicador de qualidade I_H no conjunto com $W = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.197	0.302	0.295	0.229	0.302	0.364	0.314	0.275	0.279	0.353	0.272	0.311	0.303
2	0.177	0.246	0.400	0.288	0.343	0.404	0.309	0.308	0.259	0.571	0.506	0.310	0.291
3	0.151	0.239	0.276	0.234	0.277	0.382	0.306	0.269	0.222	0.403	0.339	0.292	0.288
4	0.195	0.307	0.308	0.273	0.333	0.385	0.321	0.297	0.260	0.406	0.359	0.306	0.306
5	0.172	0.286	0.326	0.314	0.340	0.360	0.306	0.279	0.243	0.444	0.449	0.284	0.286
6	0.171	0.273	0.355	0.266	0.311	0.384	0.292	0.274	0.253	0.429	0.359	0.280	0.277
7	0.154	0.242	0.401	0.303	0.287	0.365	0.277	0.266	0.244	0.457	0.395	0.268	0.263
8	0.374	0.418	0.245	0.202	0.428	0.558	0.527	0.461	0.454	0.422	0.317	0.505	0.510
9	0.213	0.268	0.349	0.225	0.355	0.514	0.400	0.343	0.325	0.440	0.291	0.391	0.373
10	0.171	0.261	0.271	0.244	0.305	0.309	0.280	0.264	0.241	0.398	0.299	0.263	0.261
11	0.183	0.277	0.285	0.212	0.344	0.383	0.298	0.277	0.260	0.406	0.333	0.286	0.287
12	0.746	1.130	0.244	0.199	0.759	0.834	0.744	0.777	1.132	0.361	0.287	0.762	0.737
13	0.164	0.274	0.310	0.186	0.293	0.348	0.288	0.247	0.255	0.377	0.304	0.266	0.266
14	0.233	0.297	0.368	0.322	0.381	0.432	0.341	0.329	0.301	0.477	0.437	0.324	0.314
15	0.200	0.224	0.387	0.213	0.298	0.477	0.381	0.342	0.329	0.470	0.345	0.376	0.371
16	0.168	0.244	0.312	0.284	0.284	0.321	0.275	0.246	0.238	0.417	0.410	0.262	0.257
17	0.185	0.224	0.581	0.305	0.342	0.459	0.333	0.346	0.320	0.696	0.494	0.336	0.323
18	0.165	0.244	0.356	0.248	0.283	0.351	0.298	0.295	0.268	0.478	0.357	0.284	0.286
19	0.166	0.249	0.323	0.292	0.316	0.341	0.302	0.249	0.232	0.413	0.361	0.273	0.270
20	0.172	0.264	0.342	0.294	0.297	0.318	0.270	0.222	0.214	0.383	0.351	0.255	0.256

Figura 87 – *Posthoc* de Shaffer para I_H no conjunto com $W = 40$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 96 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.125	1.195	1.260	1.161	1.158	1.280	1.232	1.208	1.196	1.270	1.200	1.232	1.208
2	1.113	1.162	1.404	1.269	1.212	1.276	1.203	1.199	1.178	1.500	1.423	1.205	1.179
3	1.107	1.178	1.245	1.139	1.184	1.232	1.179	1.179	1.154	1.314	1.216	1.179	1.161
4	1.125	1.218	1.264	1.189	1.199	1.250	1.208	1.206	1.195	1.283	1.264	1.204	1.200
5	1.091	1.210	1.294	1.235	1.216	1.222	1.200	1.189	1.160	1.353	1.373	1.168	1.177
6	1.102	1.200	1.340	1.220	1.201	1.239	1.169	1.168	1.167	1.360	1.290	1.160	1.160
7	1.094	1.173	1.385	1.269	1.192	1.215	1.160	1.160	1.160	1.385	1.308	1.160	1.160
8	1.270	1.270	1.162	1.129	1.270	1.333	1.270	1.270	1.270	1.270	1.182	1.296	1.286
9	1.133	1.164	1.268	1.122	1.187	1.316	1.241	1.214	1.200	1.317	1.195	1.245	1.214
10	1.097	1.196	1.245	1.184	1.162	1.232	1.193	1.190	1.167	1.306	1.265	1.194	1.181
11	1.143	1.200	1.269	1.155	1.212	1.259	1.214	1.214	1.184	1.308	1.250	1.214	1.207
12	1.488	1.861	1.190	1.135	1.512	1.576	1.477	1.535	1.861	1.214	1.186	1.523	1.442
13	1.127	1.219	1.280	1.150	1.164	1.255	1.218	1.200	1.182	1.280	1.220	1.200	1.200
14	1.138	1.204	1.354	1.271	1.211	1.308	1.235	1.235	1.207	1.417	1.375	1.207	1.200
15	1.140	1.143	1.400	1.178	1.161	1.316	1.241	1.222	1.214	1.411	1.244	1.259	1.239
16	1.094	1.174	1.245	1.204	1.163	1.222	1.174	1.160	1.160	1.306	1.327	1.164	1.160
17	1.105	1.143	1.604	1.333	1.249	1.261	1.179	1.206	1.188	1.656	1.438	1.179	1.179
18	1.115	1.179	1.333	1.167	1.163	1.259	1.192	1.192	1.189	1.375	1.250	1.194	1.189
19	1.096	1.175	1.277	1.213	1.191	1.231	1.192	1.167	1.164	1.319	1.255	1.176	1.170
20	1.090	1.203	1.318	1.236	1.164	1.206	1.164	1.143	1.143	1.327	1.309	1.149	1.149

Figura 88 – *Posthoc* de Shaffer para I_ϵ no conjunto com $W = 40$.

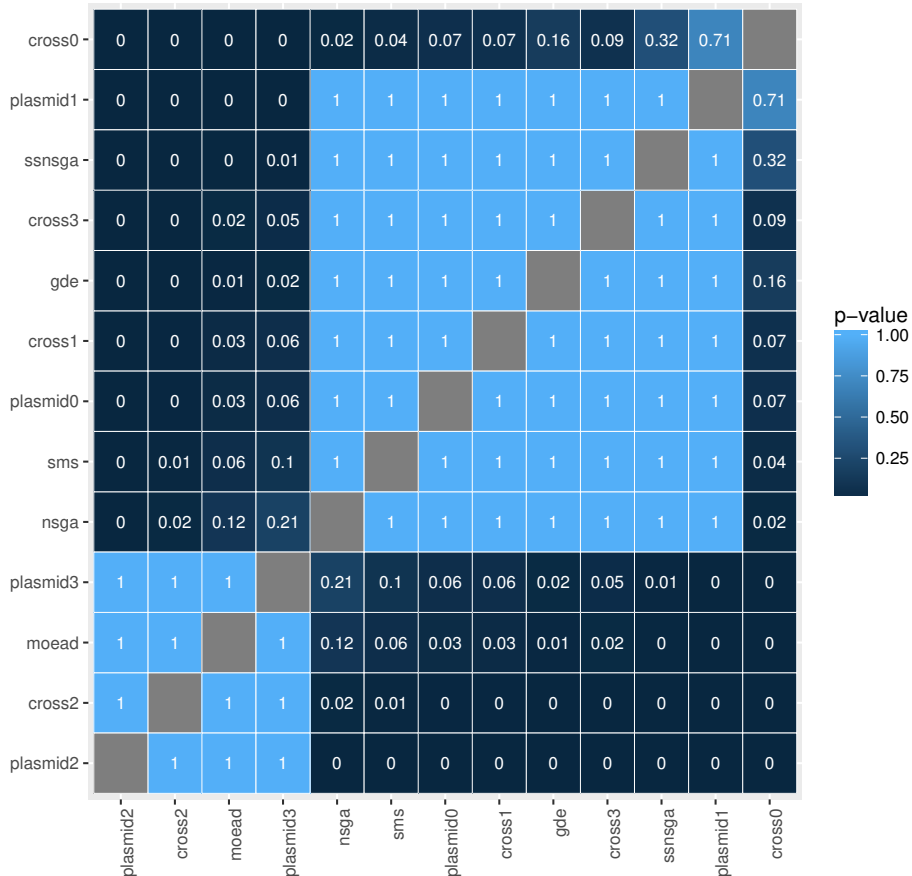
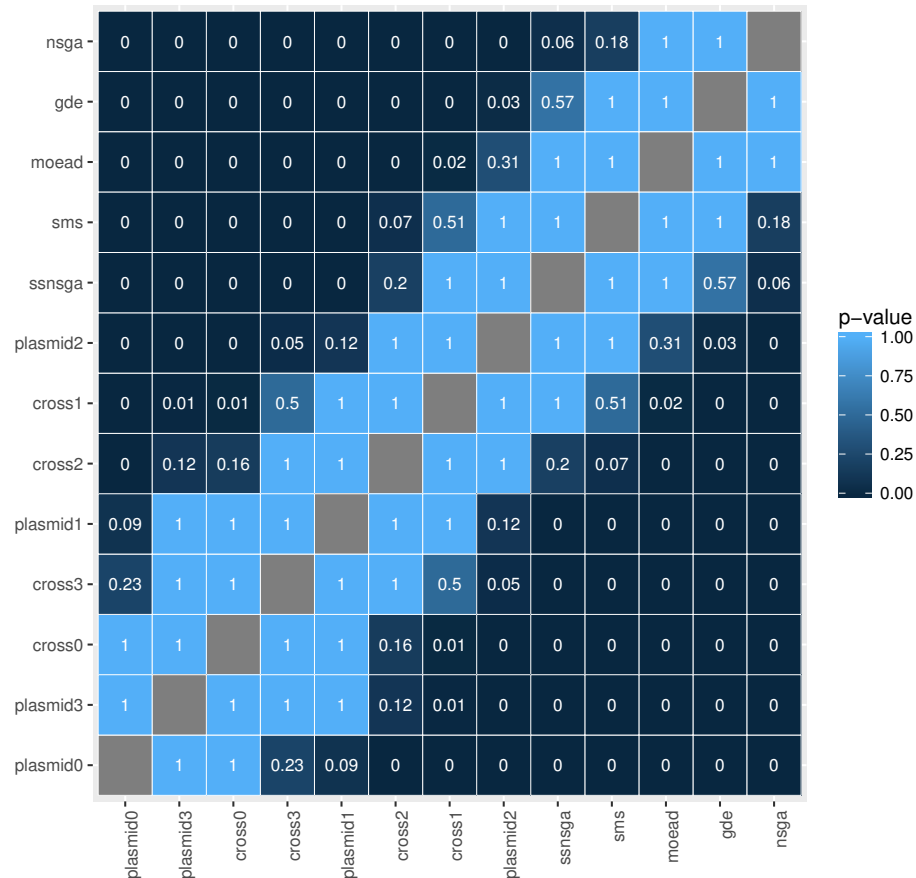


Tabela 97 – Medianas do tempo de processamento no conjunto com $W = 40$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	8.095	5.763	6.683	7.796	2.422	2.693	1.079	9.912	7.687	4.645	7.766	3.537	3.575
2	8.001	5.741	6.525	7.790	2.345	2.659	1.094	9.606	7.686	4.616	7.989	3.548	3.596
3	7.948	5.790	6.581	7.809	2.445	2.628	1.070	9.430	7.668	4.579	7.992	3.527	3.588
4	8.118	5.795	6.666	7.909	2.393	2.761	1.131	9.756	7.751	4.690	8.037	3.581	3.639
5	8.089	5.795	6.691	7.923	2.404	2.729	1.147	9.961	7.667	4.714	8.407	3.583	3.641
6	7.972	5.739	6.536	7.771	2.379	2.658	1.107	9.757	7.656	4.806	8.220	3.536	3.599
7	7.987	5.732	6.644	7.787	2.399	2.665	1.083	9.665	7.755	4.717	8.546	3.516	3.564
8	7.578	5.406	6.219	7.456	2.232	2.559	1.005	9.231	7.366	4.308	7.402	3.366	3.416
9	7.888	5.637	6.489	7.668	2.305	2.641	1.056	9.817	7.674	4.525	7.807	3.462	3.481
10	8.118	5.824	6.645	7.729	2.340	2.698	1.083	9.987	7.854	4.630	8.118	3.554	3.618
11	8.032	5.758	6.591	7.732	2.379	2.614	1.068	9.656	7.687	4.570	8.105	3.558	3.602
12	8.871	7.346	6.690	7.848	2.417	2.750	1.225	10.251	9.407	4.743	7.603	3.652	3.693
13	8.105	5.816	6.715	7.759	2.418	2.721	1.108	9.842	7.767	4.662	8.362	3.578	3.615
14	8.071	5.803	6.595	7.711	2.371	2.703	1.136	9.772	7.688	4.550	8.105	3.582	3.646
15	8.072	5.741	6.581	7.951	2.342	2.657	1.049	9.903	7.803	4.643	8.174	3.487	3.532
16	7.975	5.706	6.525	7.892	2.365	2.659	1.062	9.617	7.482	4.625	8.859	3.498	3.555
17	7.970	5.759	6.493	7.737	2.398	2.679	1.094	9.649	7.857	4.600	8.151	3.541	3.576
18	8.136	5.738	6.827	7.827	2.426	2.721	1.073	9.940	7.810	4.734	8.294	3.524	3.574
19	7.836	5.640	6.458	7.771	2.313	2.634	1.077	9.472	7.496	4.574	7.994	3.490	3.536
20	8.045	5.787	6.698	7.746	2.383	2.734	1.141	9.830	7.643	4.644	8.008	3.592	3.626

Figura 89 – *Posthoc* de Shaffer para o Tempo no conjunto com $W = 40$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 98 – Medianas do indicador de qualidade I_{ϵ_+} no conjunto com $W = 50$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.208	0.174	0.302	0.124	0.254	0.375	0.285	0.292	0.256	0.326	0.256	0.292	0.253
2	0.122	0.181	0.319	0.213	0.192	0.255	0.192	0.192	0.162	0.340	0.277	0.192	0.192
3	0.130	0.185	0.500	0.293	0.267	0.333	0.222	0.222	0.180	0.457	0.478	0.222	0.222
4	0.172	0.195	0.400	0.289	0.289	0.328	0.241	0.242	0.241	0.489	0.367	0.233	0.207
5	0.140	0.208	0.395	0.209	0.279	0.333	0.250	0.250	0.208	0.488	0.360	0.250	0.233
6	0.122	0.212	0.294	0.216	0.216	0.320	0.240	0.216	0.176	0.333	0.284	0.240	0.209
7	0.163	0.199	0.407	0.279	0.269	0.302	0.250	0.250	0.209	0.419	0.372	0.233	0.209
8	0.209	0.154	0.512	0.233	0.302	0.336	0.256	0.305	0.269	0.558	0.349	0.269	0.233
9	0.208	0.167	0.556	0.250	0.278	0.365	0.306	0.269	0.269	0.583	0.361	0.321	0.278
10	0.160	0.187	0.419	0.267	0.267	0.323	0.240	0.240	0.200	0.395	0.326	0.240	0.209
11	0.111	0.200	0.326	0.196	0.239	0.296	0.220	0.200	0.185	0.413	0.326	0.196	0.185
12	0.136	0.172	0.364	0.205	0.234	0.280	0.240	0.205	0.200	0.386	0.295	0.240	0.227
13	0.115	0.187	0.341	0.284	0.294	0.269	0.227	0.192	0.159	0.386	0.364	0.205	0.192
14	0.163	0.216	0.372	0.279	0.279	0.333	0.250	0.256	0.233	0.465	0.372	0.250	0.233
15	0.112	0.158	0.693	0.432	0.295	0.296	0.185	0.205	0.185	0.773	0.591	0.190	0.185
16	0.209	0.198	0.349	0.164	0.302	0.323	0.258	0.279	0.258	0.395	0.337	0.279	0.258
17	0.151	0.135	0.413	0.174	0.208	0.269	0.231	0.231	0.196	0.391	0.283	0.235	0.224
18	0.217	0.187	0.195	0.128	0.259	0.391	0.304	0.299	0.268	0.317	0.220	0.329	0.304
19	0.185	0.154	0.463	0.171	0.260	0.333	0.259	0.259	0.222	0.415	0.378	0.272	0.268

Figura 90 – *Posthoc* de Shaffer para I_{ϵ_+} no conjunto com $W = 50$.

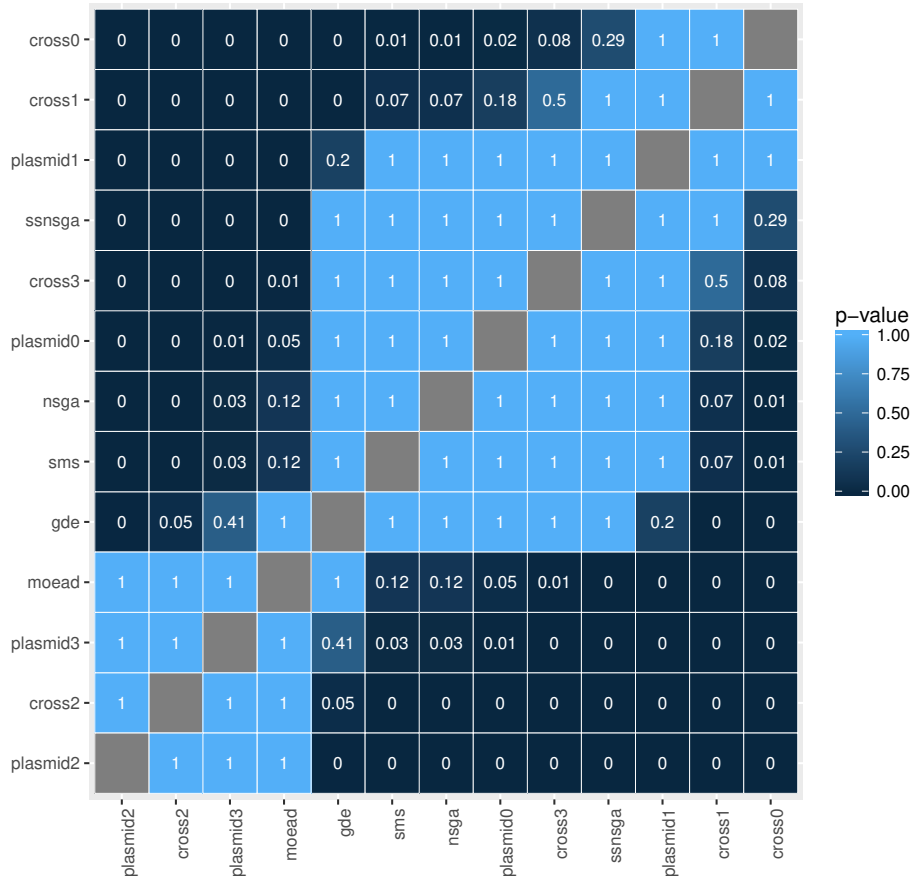
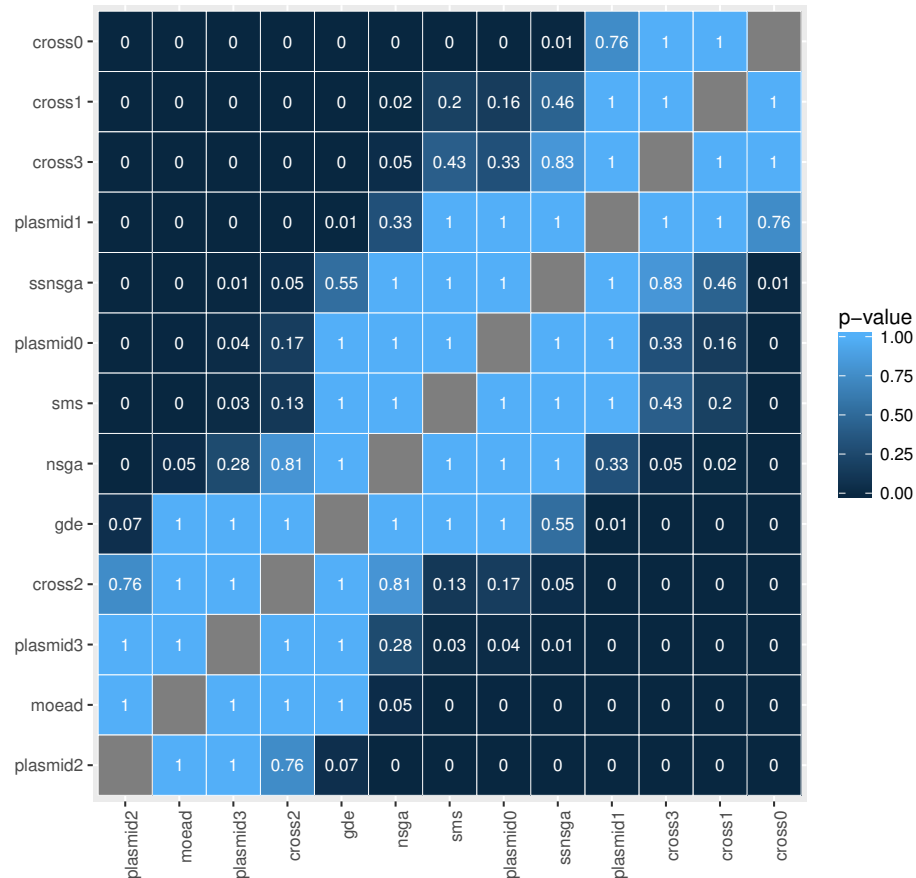


Tabela 99 – Medianas do indicador de qualidade I_H no conjunto com $W = 50$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	0.235	0.293	0.358	0.206	0.383	0.479	0.371	0.383	0.346	0.505	0.393	0.356	0.346
2	0.183	0.263	0.369	0.300	0.311	0.333	0.292	0.269	0.241	0.445	0.371	0.284	0.275
3	0.179	0.230	0.484	0.302	0.359	0.447	0.349	0.301	0.251	0.510	0.515	0.319	0.313
4	0.230	0.254	0.377	0.283	0.438	0.534	0.362	0.382	0.340	0.530	0.398	0.359	0.335
5	0.188	0.285	0.422	0.283	0.379	0.412	0.331	0.313	0.284	0.590	0.455	0.308	0.310
6	0.197	0.287	0.322	0.268	0.367	0.439	0.333	0.286	0.256	0.393	0.317	0.324	0.318
7	0.200	0.249	0.403	0.291	0.413	0.405	0.315	0.291	0.270	0.454	0.394	0.294	0.288
8	0.246	0.243	0.513	0.253	0.396	0.492	0.355	0.405	0.352	0.644	0.475	0.352	0.352
9	0.264	0.281	0.533	0.224	0.400	0.496	0.430	0.411	0.360	0.623	0.435	0.431	0.411
10	0.206	0.266	0.413	0.298	0.384	0.493	0.356	0.336	0.304	0.453	0.418	0.339	0.343
11	0.185	0.274	0.371	0.259	0.374	0.384	0.307	0.262	0.255	0.481	0.393	0.290	0.274
12	0.202	0.259	0.406	0.247	0.350	0.368	0.326	0.293	0.283	0.476	0.385	0.309	0.303
13	0.188	0.265	0.360	0.328	0.385	0.363	0.309	0.283	0.245	0.449	0.428	0.292	0.296
14	0.201	0.284	0.365	0.292	0.398	0.422	0.342	0.321	0.277	0.543	0.445	0.306	0.317
15	0.154	0.194	0.648	0.399	0.399	0.440	0.294	0.319	0.267	0.792	0.590	0.301	0.287
16	0.272	0.335	0.397	0.260	0.454	0.537	0.426	0.427	0.358	0.526	0.443	0.412	0.405
17	0.160	0.214	0.376	0.178	0.271	0.374	0.318	0.333	0.291	0.505	0.363	0.307	0.298
18	0.344	0.398	0.315	0.216	0.421	0.551	0.499	0.468	0.465	0.505	0.397	0.503	0.491
19	0.199	0.261	0.512	0.231	0.368	0.516	0.380	0.390	0.296	0.572	0.527	0.371	0.368

Figura 91 – *Posthoc* de Shaffer para I_H no conjunto com $W = 50$.



C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 100 – Medianas do indicador de qualidade I_ϵ no conjunto com $W = 50$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	1.208	1.171	1.302	1.124	1.239	1.375	1.269	1.277	1.250	1.326	1.256	1.280	1.250
2	1.112	1.181	1.319	1.213	1.187	1.226	1.189	1.179	1.154	1.340	1.277	1.182	1.179
3	1.107	1.185	1.500	1.293	1.253	1.321	1.214	1.193	1.153	1.457	1.478	1.214	1.212
4	1.155	1.195	1.400	1.289	1.275	1.305	1.212	1.233	1.228	1.489	1.367	1.208	1.198
5	1.130	1.208	1.395	1.209	1.265	1.304	1.240	1.220	1.204	1.488	1.360	1.220	1.208
6	1.120	1.212	1.294	1.216	1.209	1.300	1.213	1.200	1.160	1.333	1.284	1.213	1.200
7	1.135	1.199	1.407	1.279	1.236	1.280	1.235	1.212	1.202	1.419	1.372	1.200	1.200
8	1.185	1.148	1.512	1.233	1.273	1.298	1.222	1.262	1.229	1.558	1.349	1.231	1.222
9	1.186	1.167	1.556	1.250	1.243	1.333	1.269	1.259	1.235	1.583	1.361	1.296	1.259
10	1.130	1.187	1.419	1.267	1.256	1.308	1.222	1.216	1.189	1.395	1.326	1.222	1.192
11	1.107	1.200	1.326	1.196	1.217	1.283	1.198	1.179	1.179	1.413	1.326	1.183	1.179
12	1.120	1.172	1.364	1.205	1.227	1.269	1.218	1.192	1.172	1.386	1.295	1.218	1.200
13	1.110	1.187	1.341	1.284	1.273	1.234	1.189	1.182	1.154	1.386	1.364	1.185	1.185
14	1.137	1.216	1.372	1.279	1.256	1.293	1.250	1.250	1.208	1.465	1.372	1.214	1.208
15	1.103	1.158	1.693	1.432	1.287	1.276	1.179	1.179	1.172	1.773	1.591	1.177	1.169
16	1.200	1.193	1.349	1.164	1.289	1.323	1.250	1.257	1.250	1.395	1.337	1.258	1.250
17	1.135	1.135	1.413	1.174	1.192	1.269	1.216	1.219	1.192	1.391	1.283	1.231	1.204
18	1.208	1.187	1.195	1.128	1.244	1.372	1.292	1.286	1.262	1.317	1.220	1.313	1.304
19	1.174	1.153	1.463	1.171	1.239	1.333	1.250	1.250	1.211	1.415	1.378	1.250	1.250

Figura 92 – *Posthoc* de Shaffer para I_ϵ no conjunto com $W = 50$.

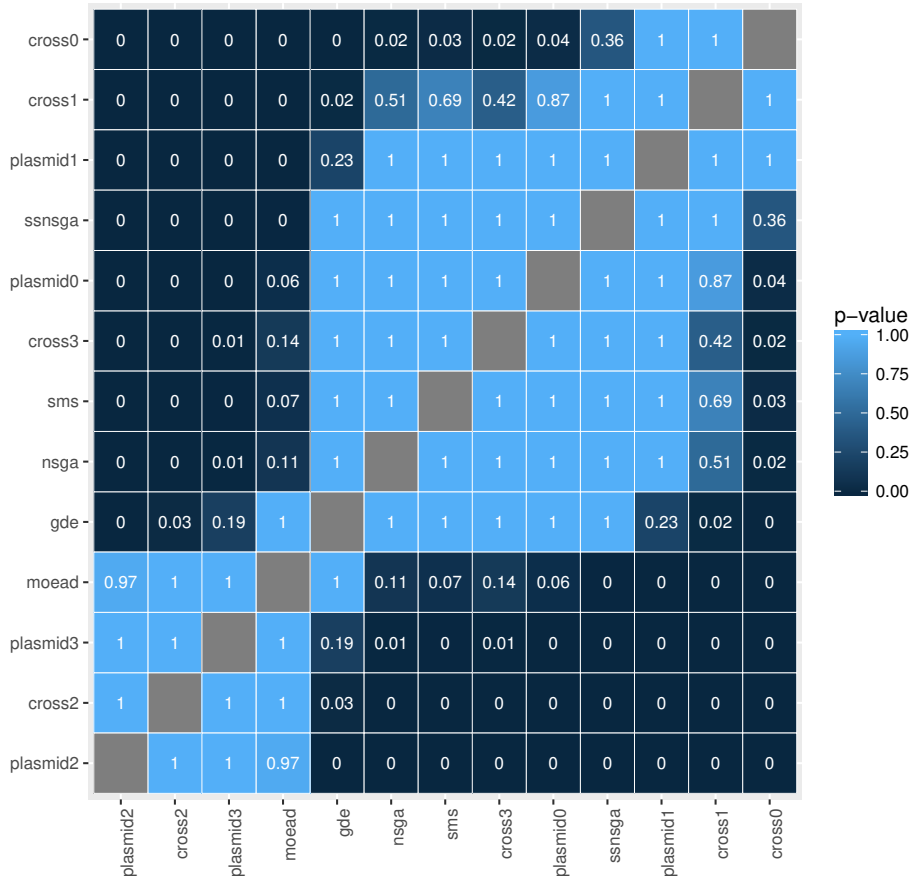


Tabela 101 – Medianas do tempo de processamento no conjunto com $W = 50$.

Instância	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga
1	10.211	7.245	8.420	9.945	2.872	3.358	1.342	12.251	9.711	5.868	10.330	4.495	4.551
2	9.831	7.067	8.000	9.745	2.838	3.230	1.333	11.858	9.332	5.676	10.068	4.399	4.490
3	10.151	7.246	8.371	9.868	2.884	3.431	1.421	12.246	9.717	5.761	10.209	4.509	4.570
4	10.295	7.325	8.480	10.046	2.848	3.417	1.448	12.325	9.964	5.825	10.391	4.535	4.611
5	10.034	7.183	8.331	9.790	2.900	3.372	1.419	12.515	9.632	5.702	10.667	4.478	4.581
6	10.331	7.361	8.608	9.959	2.913	3.508	1.471	12.516	9.910	6.046	10.429	4.546	4.622
7	9.973	7.128	8.212	9.534	3.006	3.270	1.361	12.031	9.443	5.600	9.348	4.449	4.507
8	10.155	7.223	8.417	9.830	3.014	3.193	1.316	12.387	9.702	5.803	10.347	4.446	4.520
9	9.889	7.078	8.078	9.554	3.018	3.231	1.302	12.009	9.473	5.679	10.216	4.375	4.446
10	10.051	7.222	8.242	9.696	2.993	3.351	1.385	12.442	9.690	5.790	10.583	4.468	4.569
11	9.919	7.146	8.127	9.567	3.052	3.252	1.385	12.004	9.339	5.582	10.030	4.461	4.519
12	9.699	6.954	8.039	9.597	2.968	3.216	1.272	11.718	9.141	5.569	9.429	4.316	4.377
13	9.891	7.120	8.029	9.803	3.044	3.228	1.344	11.958	9.369	5.561	10.078	4.438	4.521
14	10.105	7.215	8.353	9.876	3.047	3.368	1.385	12.710	9.606	6.094	10.411	4.510	4.582
15	9.947	7.194	8.254	9.689	3.004	3.243	1.338	11.978	9.598	5.781	9.621	4.421	4.515
16	10.172	7.295	8.295	9.748	2.955	3.386	1.410	12.500	9.679	5.708	9.712	4.537	4.594
17	9.924	6.986	8.212	9.641	2.955	3.302	1.329	11.928	9.543	5.649	10.035	4.320	4.388
18	9.686	6.844	7.945	9.497	2.740	3.123	1.233	11.600	9.343	5.599	9.538	4.282	4.335
19	10.093	7.198	8.336	9.903	2.803	3.310	1.367	12.447	9.539	5.899	10.520	4.482	4.526

Figura 93 – *Posthoc* de Shaffer para o Tempo no conjunto com $W = 50$.

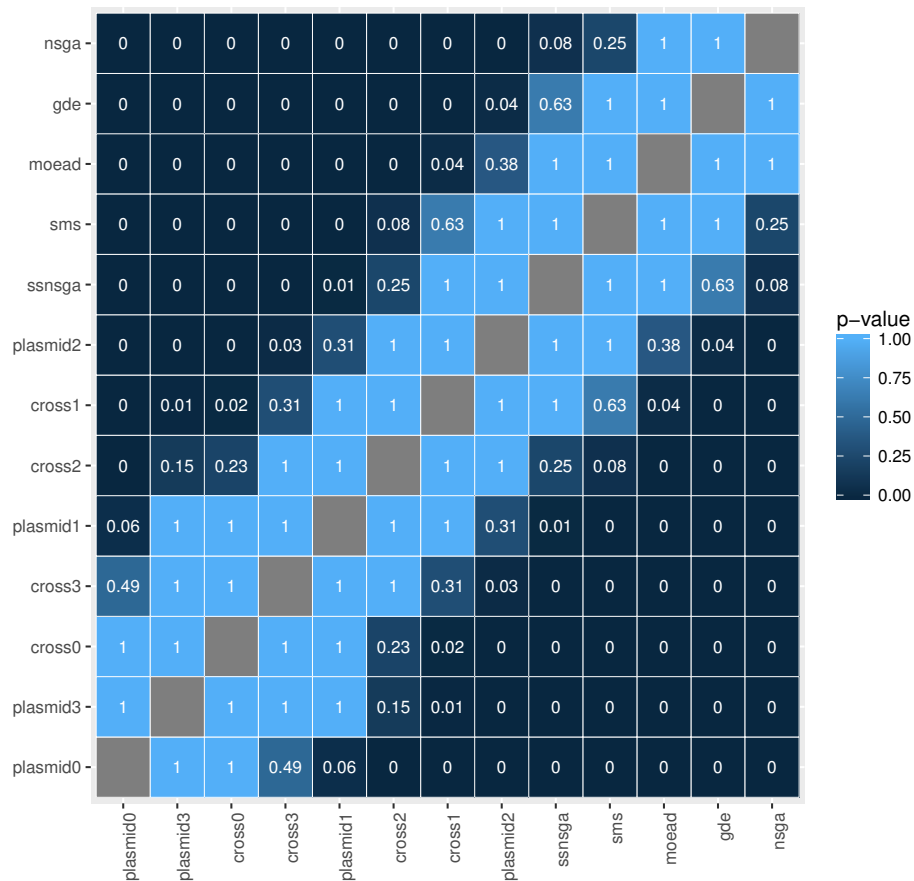


Tabela 102 – Contagem de resultados em relação ao indicador I_{ϵ_+} no tamanho da rede(n).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	3	4	5	0	5	5	1	5	5	4	5	5	78.3
cross1	0	0	1	0	0	4	1	0	0	3	0	1	1	18.3
cross2	0	1	0	1	0	3	1	0	0	2	0	1	1	16.7
cross3	0	0	0	0	0	4	0	0	0	3	0	0	0	11.7
gde	0	1	2	0	0	5	5	0	1	5	0	4	1	40
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid0	0	1	1	0	0	5	3	0	0	4	0	0	0	23.3
plasmid1	0	1	0	0	0	4	0	0	0	1	0	0	0	10
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	1	0	1	0	5	1	0	0	4	0	1	1	23.3
sms	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ssnsga	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Derrotas	0	13.3	13.3	11.7	0	58.3	26.7	1.7	10	45	6.7	20	15	

Tabela 103 – Contagem de resultados em relação ao indicador I_H no tamanho da rede(n).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	3	4	5	0	5	5	0	5	5	5	5	5	78.3
cross1	0	0	0	0	0	5	3	0	0	4	0	0	0	20
cross2	0	0	0	0	0	4	1	0	0	4	0	1	1	18.3
cross3	0	0	0	0	0	4	1	0	0	4	0	0	0	15
gde	0	0	1	0	0	5	5	0	2	5	3	1	1	38.3
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid0	0	0	1	2	0	5	5	0	2	5	4	4	4	53.3
plasmid1	0	0	0	0	0	3	0	0	0	2	0	0	0	8.3
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	0	0	0	0	1	1	0	0	2	0	0	0	6.7
sms	0	0	0	0	0	0	0	0	0	3	0	0	0	5
ssnsga	0	0	0	0	0	1	0	0	0	3	0	0	0	6.7
Derrotas	0	5	10	11.7	0	55	35	0	15	61.7	20	18.3	18.3	

Tabela 104 – Contagem de resultados em relação ao indicador I_c no tamanho da rede (n) .

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	4	4	5	1	5	5	0	5	5	4	5	5	80
cross1	0	0	1	0	0	4	1	0	0	3	0	0	0	15
cross2	0	1	0	1	0	2	1	0	0	1	0	0	0	10
cross3	0	0	0	0	0	2	0	0	0	2	0	0	0	6.7
gde	0	1	2	0	0	5	4	0	1	5	1	3	1	38.3
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid0	0	1	1	1	0	5	2	0	0	4	0	0	0	23.3
plasmid1	0	1	0	0	0	3	0	0	0	1	0	0	0	8.3
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	1	0	1	0	3	1	0	0	3	0	0	0	15
sms	0	1	0	0	0	0	0	0	0	0	0	0	0	1.7
ssnsga	0	1	0	0	0	1	0	0	0	1	0	0	0	5
Derrotas	0	18.3	13.3	13.3	1.7	50	23.3	0	10	41.7	8.3	13.3	10	

Tabela 105 – Contagem de resultados em relação ao Tempo de processamento no tamanho da rede (n) .

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cross1	5	0	0	4	0	0	0	5	0	0	0	0	0	23.3
cross2	0	0	0	1	0	0	0	5	0	0	0	0	0	10
cross3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gde	5	5	5	5	0	0	0	5	5	2	5	0	0	61.7
moead	5	5	5	5	0	0	0	5	5	1	5	0	0	60
nsga	5	5	5	5	0	0	0	5	5	5	5	0	0	66.7
plasmid0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid2	5	0	0	5	0	0	0	5	2	0	4	0	0	35
plasmid3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sms	5	0	0	5	0	0	0	5	5	0	5	0	0	41.7
ssnsga	5	0	0	5	0	0	0	5	5	0	5	0	0	41.7
Derrotas	58.3	25	25	58.3	0	0	0	66.7	45	13.3	48.3	0	0	

Tabela 106 – Contagem de resultados em relação ao indicador I_{ϵ_+} no número de participantes (D^k) .

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	5	2	3	0	5	3	3	5	5	2	2	2	61.7
cross1	0	0	0	0	0	1	0	0	0	2	0	0	0	5
cross2	0	3	0	0	0	3	0	0	3	3	0	0	0	20
cross3	0	1	0	0	0	2	0	0	2	2	0	0	0	11.7
gde	0	3	1	0	0	5	1	0	4	4	1	1	0	33.3
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	3	0	0	0	1	0	0	3	0	0	0	0	11.7
plasmid0	0	2	0	0	0	2	0	0	3	2	0	0	0	15
plasmid1	0	0	0	0	0	1	0	0	0	1	0	0	0	3.3
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	3	0	1	0	3	0	0	3	4	0	0	0	23.3
sms	0	3	0	0	0	2	0	0	3	3	0	0	0	18.3
ssnsga	0	3	0	0	0	4	0	0	3	3	0	0	0	21.7
Derrotas	0	43.3	5	6.7	0	48.3	6.7	5	48.3	48.3	5	5	3.3	

Tabela 107 – Contagem de resultados em relação ao indicador I_H no número de participantes (D^k).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	5	2	2	0	5	5	0	5	5	4	3	3	65
cross1	0	0	0	0	0	1	0	0	0	2	0	0	0	5
cross2	0	3	0	0	0	5	3	0	4	5	2	0	1	38.3
cross3	0	2	0	0	0	4	0	0	4	3	0	0	0	21.7
gde	0	3	0	0	0	5	1	0	4	4	1	0	0	30
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	0	0	0	0	0	0	0	3	1	0	0	0	6.7
plasmid0	0	5	1	0	0	3	2	0	4	3	2	1	1	36.7
plasmid1	0	0	0	0	0	1	0	0	0	1	0	0	0	3.3
plasmid2	0	0	0	0	0	0	0	0	1	0	0	0	0	1.7
plasmid3	0	1	0	0	0	0	0	0	3	0	0	0	0	6.7
sms	0	1	0	0	0	1	0	0	3	1	0	0	0	10
ssnsga	0	1	0	0	0	1	0	0	3	2	0	0	0	11.7
Derrotas	0	35	5	3.3	0	43.3	18.3	0	56.7	45	15	6.7	8.3	

Tabela 108 – Contagem de resultados em relação ao indicador I_ϵ no número de participantes (D^k).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	5	2	5	0	5	2	2	5	5	2	2	2	61.7
cross1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cross2	0	3	0	0	0	1	0	0	3	2	0	0	0	15
cross3	0	1	0	0	0	1	0	0	2	1	0	0	0	8.3
gde	0	3	1	1	0	3	1	0	4	4	1	0	0	30
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	3	0	0	0	1	0	0	3	3	0	0	0	16.7
plasmid0	0	3	0	0	0	1	0	0	4	2	0	0	0	16.7
plasmid1	0	0	0	0	0	1	0	0	0	1	0	0	0	3.3
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	3	0	1	0	2	0	0	3	4	0	0	0	21.7
sms	0	3	0	0	0	1	0	0	4	3	0	0	0	18.3
ssnsga	0	3	0	1	0	4	0	0	4	4	0	0	0	26.7
Derrotas	0	45	5	13.3	0	33.3	5	3.3	53.3	48.3	5	3.3	3.3	

Tabela 109 – Contagem de resultados em relação ao Tempo de processamento no número de participantes (D^k).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cross1	5	0	0	0	0	0	0	5	0	0	4	0	0	23.3
cross2	0	0	0	0	0	0	0	5	0	0	0	0	0	8.3
cross3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gde	5	5	5	5	0	0	0	5	5	1	5	0	0	60
moead	5	5	5	5	0	0	0	5	5	0	5	0	0	58.3
nsga	5	5	5	5	0	0	0	5	5	5	5	0	0	66.7
plasmid0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid1	0	0	0	0	0	0	0	2	0	0	0	0	0	3.3
plasmid2	5	0	0	3	0	0	0	5	0	0	4	0	0	28.3
plasmid3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sms	5	0	1	5	0	0	0	5	5	0	5	0	0	43.3
ssnsga	5	0	0	5	0	0	0	5	5	0	5	0	0	41.7
Derrotas	58.3	25	26.7	46.7	0	0	0	70	41.7	10	55	0	0	

C. RESULTADOS DO MODELO MULTIOBJETIVO
EXPERIMENTO 1

Tabela 110 – Contagem de resultados em relação ao indicador I_{ϵ_+} no número de sessões (W).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	2	5	2	1	5	5	3	2	5	5	5	3	71.7
cross1	0	0	3	0	1	5	1	0	0	3	3	1	0	28.3
cross2	0	0	0	0	0	2	0	0	0	0	0	0	0	3.3
cross3	0	0	2	0	0	4	1	0	0	4	2	1	0	23.3
gde	0	0	3	0	0	4	3	0	0	5	1	1	0	28.3
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	0	1	0	0	1	0	0	0	2	1	0	0	8.3
plasmid0	0	0	3	0	0	5	2	0	0	4	2	1	0	28.3
plasmid1	0	0	2	0	0	5	0	0	0	4	2	0	0	21.7
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	0	0	0	0	2	0	0	0	0	0	0	0	3.3
sms	0	0	2	0	0	1	0	0	0	2	1	0	0	10
ssnsga	0	0	2	0	0	4	0	0	0	2	1	0	0	15
Derrotas	0	3.3	38.3	3.3	3.3	63.3	20	5	3.3	51.7	30	15	5	

Tabela 111 – Contagem de resultados em relação ao indicador I_H no número de sessões (W).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	2	4	2	4	5	5	1	3	5	5	5	5	76.7
cross1	0	0	2	1	1	5	2	0	0	4	3	0	0	30
cross2	0	0	0	0	0	2	1	0	0	2	0	0	0	8.3
cross3	0	0	2	0	2	4	3	0	0	4	2	0	0	28.3
gde	0	0	1	1	0	3	2	0	0	2	1	0	0	16.7
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	0	0	0	0	1	0	0	0	1	0	0	0	3.3
plasmid0	0	1	2	1	0	5	3	0	0	5	3	2	2	40
plasmid1	0	0	2	0	1	5	2	0	0	5	2	0	0	28.3
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	0	0	0	0	2	0	0	0	1	0	0	0	5
sms	0	0	0	0	0	4	0	0	0	3	1	0	0	13.3
ssnsga	0	0	1	0	0	4	0	0	0	4	1	0	0	16.7
Derrotas	0	5	23.3	8.3	13.3	66.7	30	1.7	5	60	30	11.7	11.7	

Tabela 112 – Contagem de resultados em relação ao indicador I_ϵ no número de sessões (W).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	2	5	3	1	5	5	2	2	5	5	5	3	71.7
cross1	0	0	3	0	1	3	0	0	0	3	2	0	0	20
cross2	0	0	0	0	0	1	0	0	0	0	0	0	0	1.7
cross3	0	0	2	0	0	3	0	0	0	3	2	0	0	16.7
gde	0	0	3	1	0	4	2	0	0	5	2	1	0	30
moead	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nsga	0	0	2	0	0	0	0	0	0	2	1	0	0	8.3
plasmid0	0	0	3	1	0	4	1	0	0	4	2	0	0	25
plasmid1	0	0	2	0	0	5	0	0	0	4	2	0	0	21.7
plasmid2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid3	0	0	0	0	0	2	0	0	0	0	0	0	0	3.3
sms	0	0	2	0	0	0	0	0	0	2	1	0	0	8.3
ssnsga	0	0	2	0	0	4	0	0	0	2	2	0	0	16.7
Derrotas	0	3.3	40	8.3	3.3	51.7	13.3	3.3	3.3	50	31.7	10	5	

Tabela 113 – Contagem de resultados em relação ao Tempo de processamento no número de sessões (W).

	cross0	cross1	cross2	cross3	gde	moed	nsga	plasmid0	plasmid1	plasmid2	plasmid3	sms	ssnsga	Vitorias(%)
cross0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cross1	5	0	0	1	0	0	0	5	0	0	4	0	0	25
cross2	0	0	0	0	0	0	0	5	0	0	0	0	0	8.3
cross3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gde	5	5	5	5	0	0	0	5	5	4	5	0	0	65
moead	5	5	5	5	0	0	0	5	5	0	5	0	0	58.3
nsga	5	5	5	5	0	0	0	5	5	5	5	0	0	66.7
plasmid0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasmid2	5	0	0	5	0	0	0	5	0	0	4	0	0	31.7
plasmid3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sms	5	0	0	5	0	0	0	5	5	0	5	0	0	41.7
ssnsga	5	0	0	5	0	0	0	5	5	0	5	0	0	41.7
Derrotas	58.3	25	25	51.7	0	0	0	66.7	41.7	15	55	0	0	

D Resultados do Modeo Multiobjetivo

Experimento 2

Neste apêndice, os resultados obtidos pelos algoritmos são apresentados de forma detalhada. São apresentados os resultados estatísticos das comparações entre os pares de algoritmos escolhidos para avaliação.

O conjunto de instâncias escolhido para realizar a comparação consiste de 30 instâncias. Estas instâncias foram escolhidas dos conjuntos de instâncias definido na Tabela 22, Seção 7.6.4.1. As instâncias são classificadas por número de vértices e sessões, n e W , respectivamente.

A avaliação considerou quatro métricas: I_H , $I_{\epsilon+}$, I_{ϵ} e tempo de processamento medido em segundos. A metodologia experimental foi definida na Seção 7.6.4.1.

O objetivo deste experimento é avaliar o comportamento dos algoritmos em relação ao conjunto de Pareto ótimo de cada uma das instâncias nas métricas selecionadas. As Figuras 94-213 ilustram o resultado do *PostHoc* de Nemeyi da comparação entre os algoritmos em cada uma das instâncias consideradas.

Nas Tabelas 116-155, é apresentado o número de melhores resultados que os algoritmos conseguem por grupos de instâncias (separados por W). Dois tipos de tabelas são utilizadas: o primeiro tipo considera o número máximo de vitórias e derrotas que cada algoritmo obteve no conjunto de instâncias por valor de W , conforme ilustrado na Tabela 114.

Tabela 114 – Contagem de melhores resultado no $I_{\epsilon+}$ e $W=5$.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	2	3	5	3	6	5	2	4	5	5	6	5	51
cross1	0	0	3	4	1	4	2	0	0	5	5	3	2	29
cross2	0	2	0	1	1	3	5	3	2	2	1	3	3	26
cross3	0	2	0	0	0	2	1	1	2	1	1	2	2	14
plasm0	1	2	4	2	0	4	2	0	2	4	3	2	1	27
plasm1	0	0	0	3	0	0	0	3	3	1	3	0	0	13
plasm2	0	0	0	3	1	0	0	2	2	1	3	0	0	12
plasm3	0	2	1	3	1	3	3	0	3	3	1	3	2	25
moead	0	0	2	2	0	0	0	0	0	0	2	0	0	6
sms	0	0	0	0	0	1	2	2	4	0	2	1	1	13
nsga	0	0	0	0	0	2	2	2	3	0	0	0	1	10
ssnsga	0	0	2	2	0	0	0	1	2	2	4	0	0	13
gde	0	0	2	2	1	0	0	1	2	2	3	0	0	13
Derrotas	1	10	17	27	8	25	22	17	29	26	33	20	17	

D. RESULTADOS DO MODEO MULTIOBJETIVO EXPERIMENTO 2

Na Tabela 114, cada linha ilustra o número de melhores resultados obtidos por cada algoritmo em relação aos demais no conjunto de instâncias com $W = 5$ no indicador de qualidade I_{ϵ_+} . Cada coluna ilustra o número de vezes em que cada algoritmo foi derrotado.

Tabela 115 – Contagem de melhores resultado no I_{ϵ_+} e $W=5$.

	W=5											
	b30_01		b30_02		b30_04		b60_01		b60_02		b60_03	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	9	0	9	0	7	0	10	0	10	0	6	1
cross1	8	0	5	0	7	0	2	4	1	5	6	1
cross2	0	4	2	6	2	6	9	0	7	0	6	1
cross3	1	3	0	9	0	9	5	2	7	0	1	4
plasm0	6	0	5	0	4	2	2	3	6	2	4	1
plasm1	4	3	4	3	3	2	0	7	0	6	2	4
plasm2	1	4	2	4	1	3	5	2	1	5	2	4
plasm3	5	3	1	6	1	4	5	3	5	1	8	0
moea	0	5	2	3	2	3	1	6	1	6	0	6
sms	1	4	3	6	4	4	3	3	1	5	1	4
nsga	0	4	3	6	4	4	1	7	1	5	1	7
ssnsga	1	4	5	1	4	2	1	5	0	4	2	4
gde	1	3	4	1	2	2	2	4	2	3	2	4

Na Tabela 115, o resultado é ilustrado por instância do conjunto $W = 5$. Ilustra-se o número de vitórias e derrotas obtidas por cada algoritmo. O número máximo de vitórias é 12. Por exemplo, a versão **cross0** do transgenético obteve 9 vitórias na instância b30_01 em comparação aos demais. Os resultados no restante do apêndice seguem a mesma interpretação.

Figura 94 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_01 e $W = 5$.

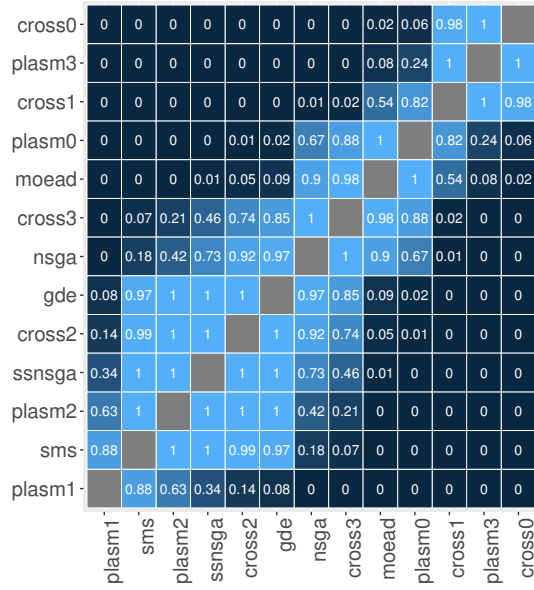


Figura 96 – *Posthoc* de Nemenyi I_H para b30_01 e $W = 5$.

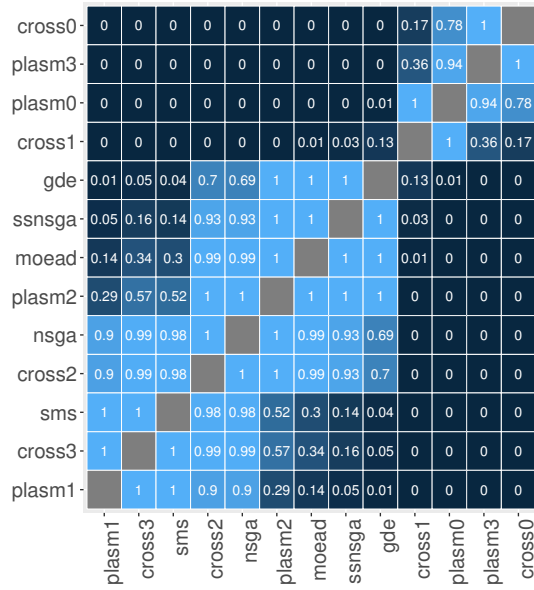


Figura 95 – *Posthoc* de Nemenyi I_{ϵ} para b30_01 e $W = 5$.

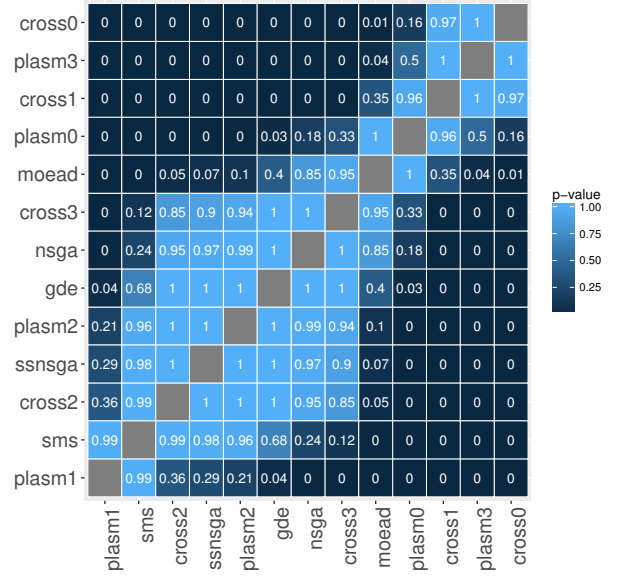


Figura 97 – *Posthoc* de Nemenyi Tempo(s) para b30_01 e $W = 5$.

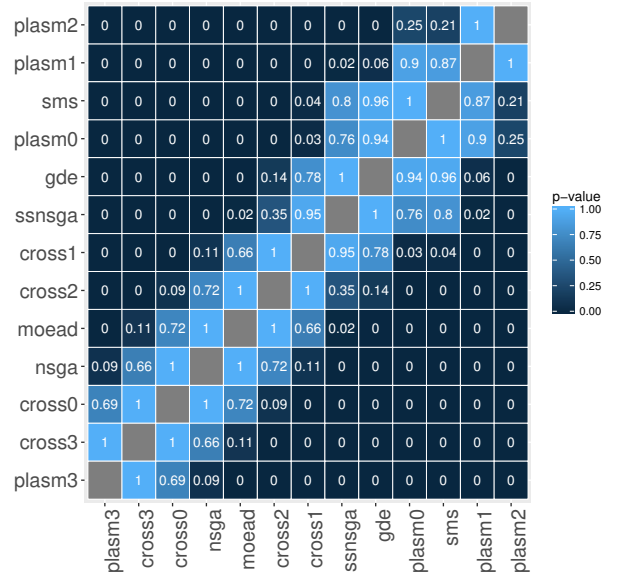


Figura 98 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_02 e $W = 5$.

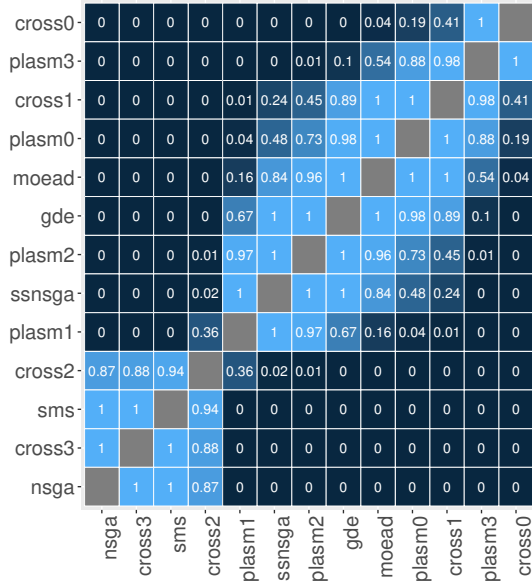


Figura 100 – *Posthoc* de Nemenyi I_H para b30_02 e $W = 5$.

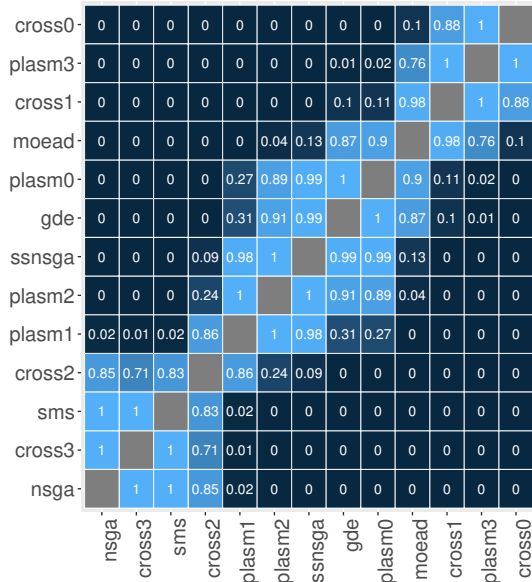


Figura 99 – *Posthoc* de Nemenyi I_{ϵ} para b30_02 e $W = 5$.

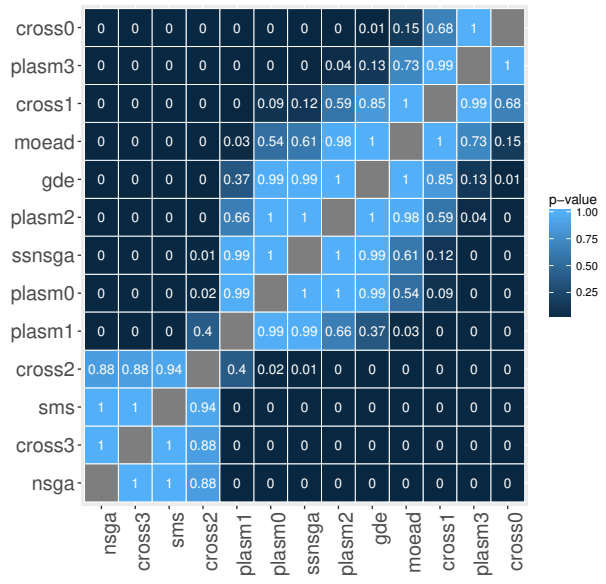


Figura 101 – *Posthoc* de Nemenyi Tempo(s) para b30_02 e $W = 5$.

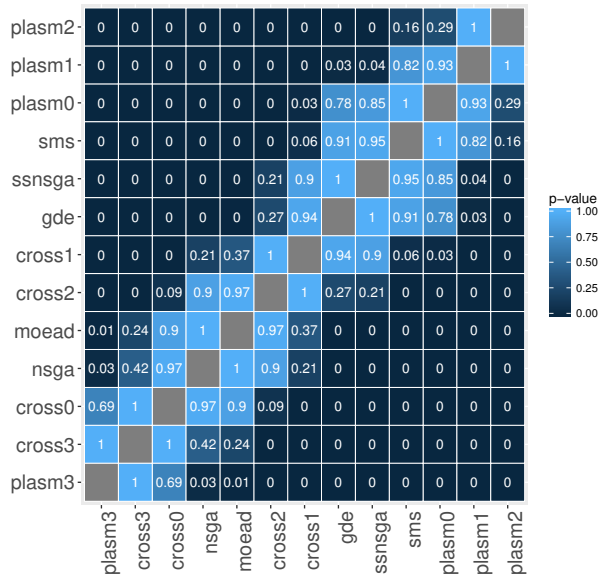


Figura 102 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_04 e $W = 5$.

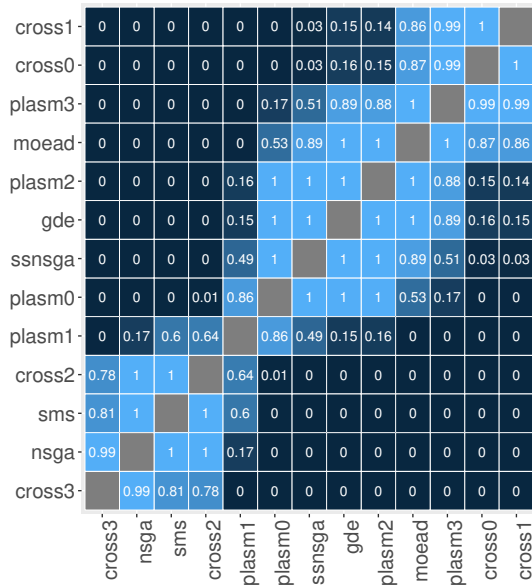


Figura 103 – *Posthoc* de Nemenyi I_{ϵ} para b30_04 e $W = 5$.

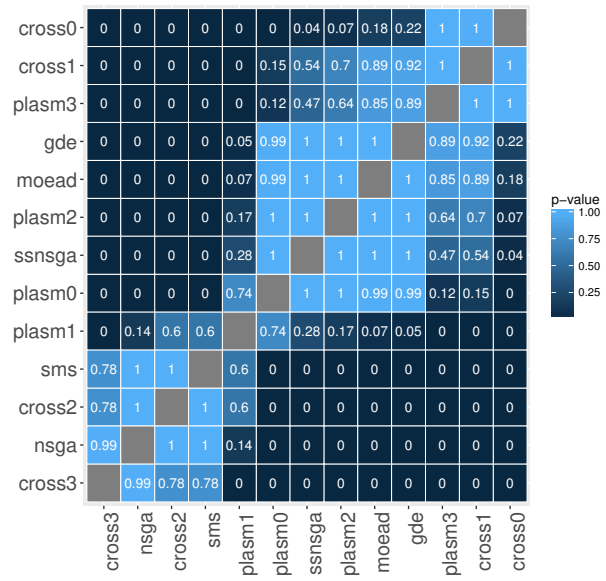


Figura 104 – *Posthoc* de Nemenyi I_H para b30_04 e $W = 5$.

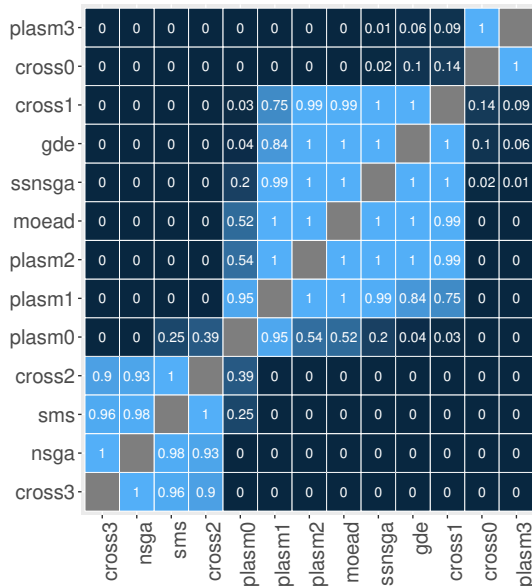


Figura 105 – *Posthoc* de Nemenyi Tempo(s) para b30_04 e $W = 5$.

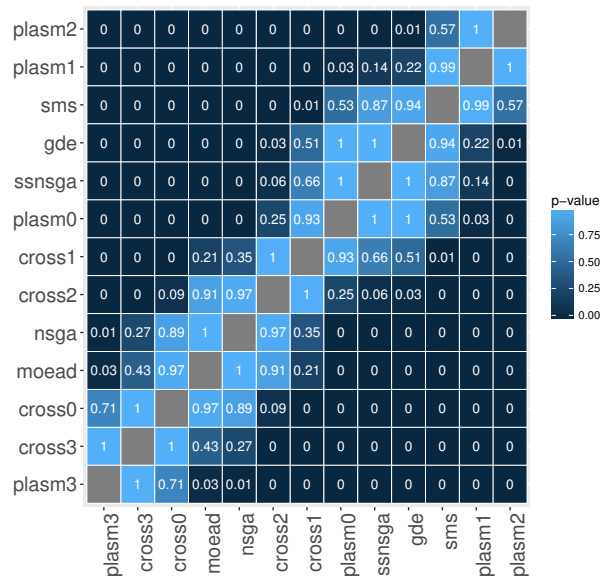


Figura 106 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_07 e $W = 10$.

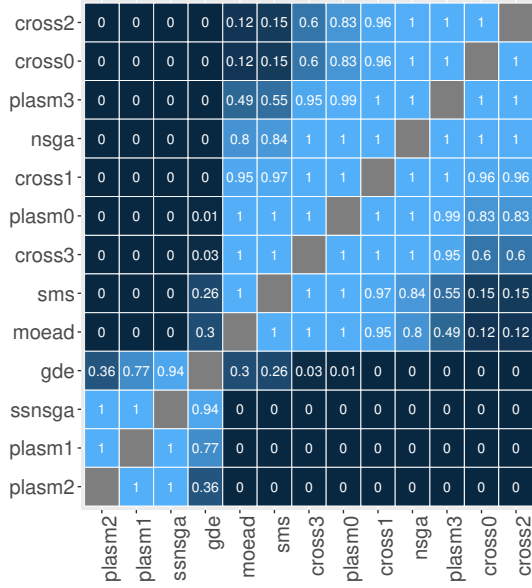


Figura 108 – *Posthoc* de Nemenyi I_H para b30_07 e $W = 10$.

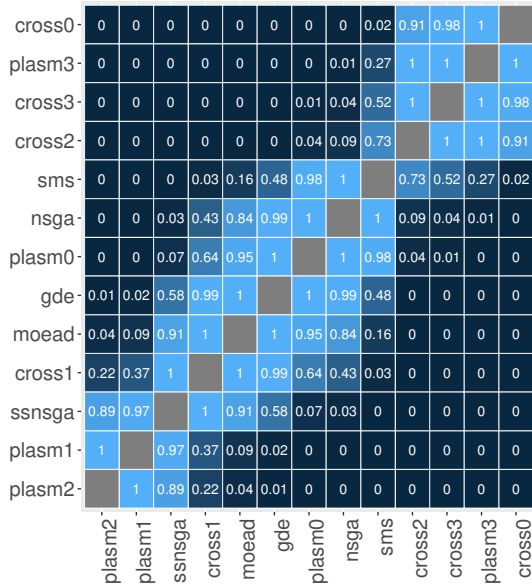


Figura 107 – *Posthoc* de Nemenyi I_{ϵ} para b30_07 e $W = 10$.

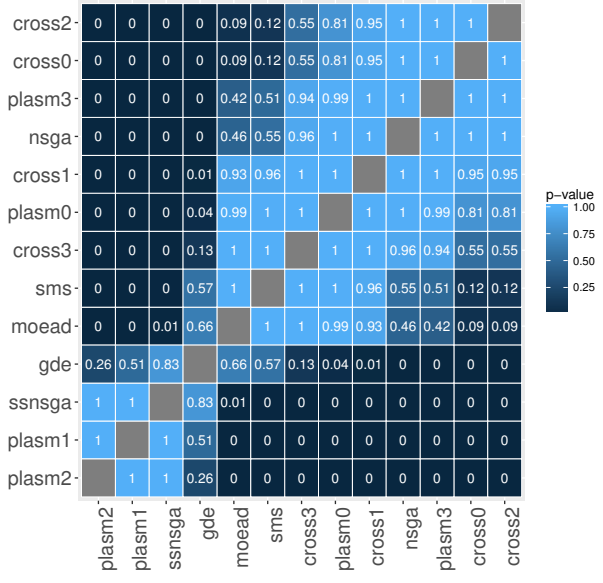


Figura 109 – *Posthoc* de Nemenyi Tempo(s) para b30_07 e $W = 10$.

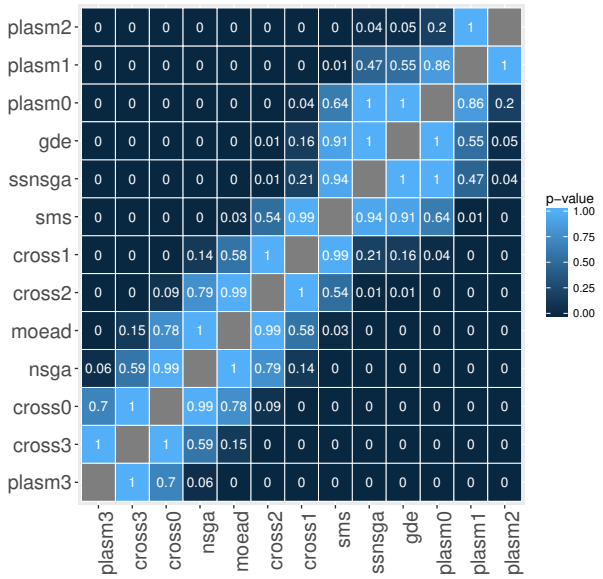


Figura 110 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_08 e $W = 10$.

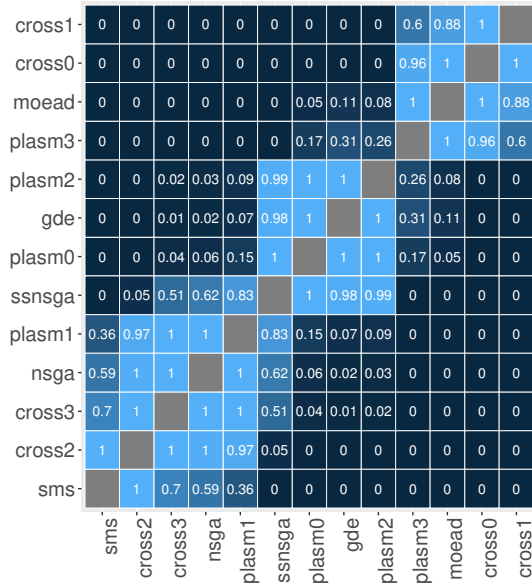


Figura 112 – *Posthoc* de Nemenyi I_H para b30_08 e $W = 10$.

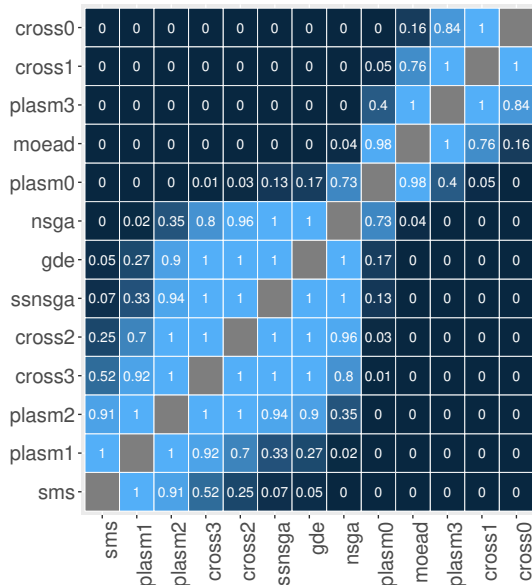


Figura 111 – *Posthoc* de Nemenyi I_{ϵ} para b30_08 e $W = 10$.

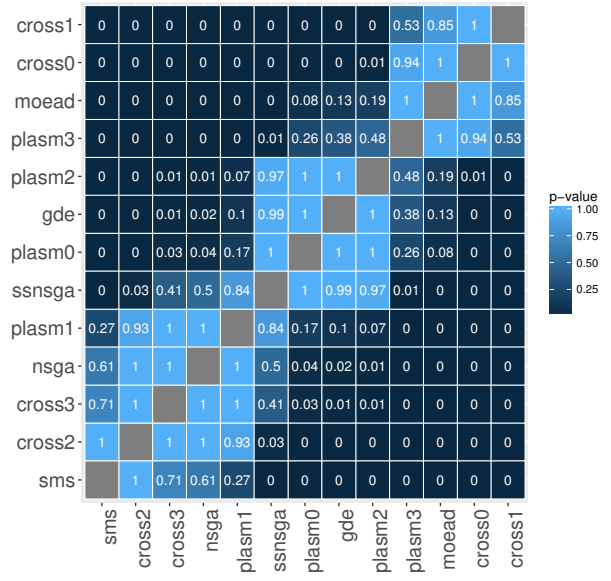
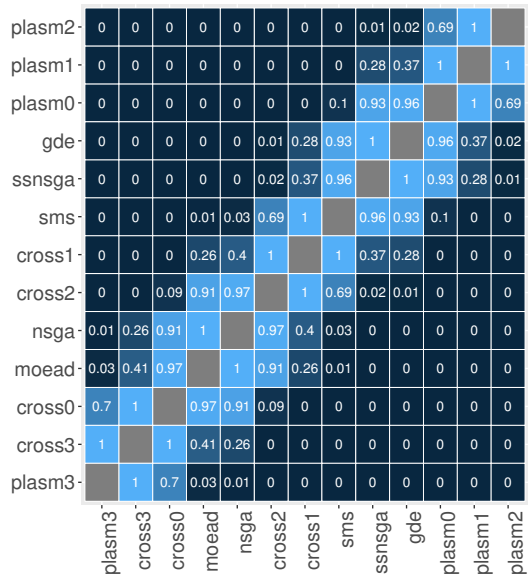


Figura 113 – *Posthoc* de Nemenyi Tempo(s) para b30_08 e $W = 10$.



D. RESULTADOS DO MODEO MULTIOBJETIVO EXPERIMENTO 2

Figura 114 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_09 e $W = 10$.

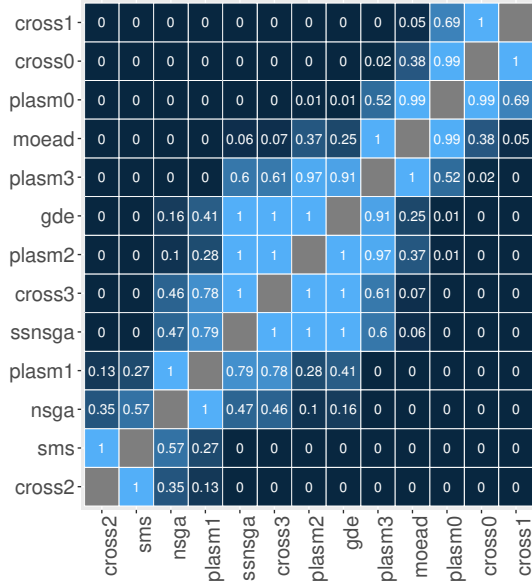


Figura 115 – *Posthoc* de Nemenyi I_{ϵ} para b30_09 e $W = 10$.

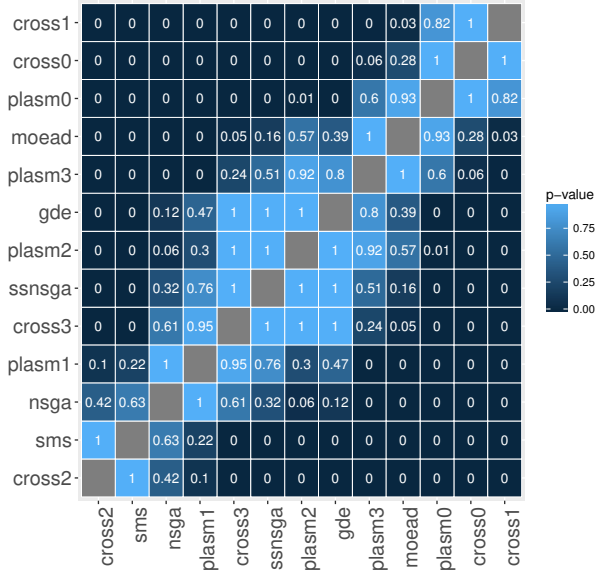


Figura 116 – *Posthoc* de Nemenyi I_H para b30_09 e $W = 10$.

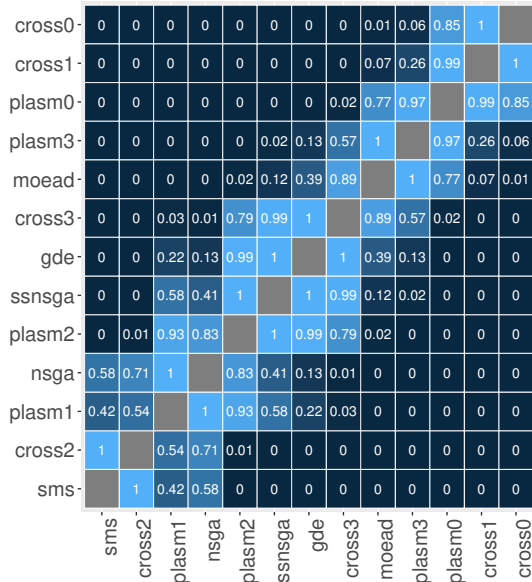
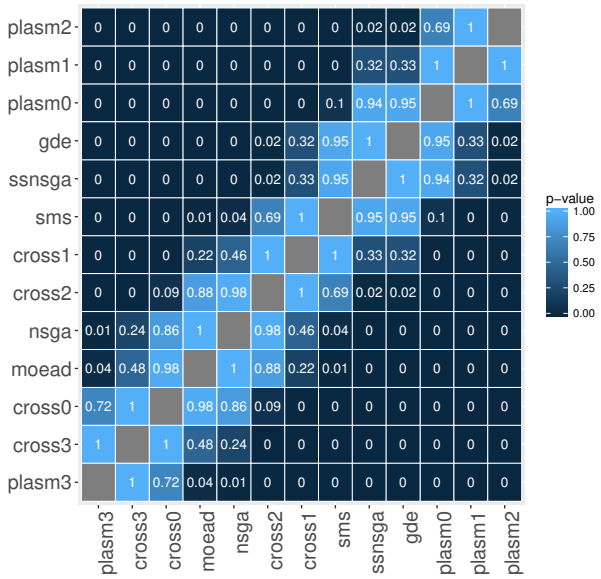


Figura 117 – *Posthoc* de Nemenyi Tempo(s) para b30_09 e $W = 10$.



cross0	0	0	0	0	0	0	0	0	0.02	0.04	0.8	0.87
cross1	0	0	0	0	0	0	0	0	0.84	0.92	1	0.87
plasm0	0	0	0	0	0	0	0	0	0.9	0.96		1
moead	0	0	0	0	0	0	0.07	0.06	1		0.96	0.92
plasm3	0	0	0	0	0	0.01	0.12	0.1		1	0.9	0.84
plasm2	0	0	0.06	1	1	1	1		0.1	0.06	0	0
gde	0	0	0.05	0.99	1	1		1	0.12	0.07	0	0
ssnsga	0	0.04	0.4	1	1		1	1	0.01	0	0	0
nsga	0	0.08	0.59	1		1	1	1	0	0	0	0
cross2	0	0.1	0.63		1	1	0.99	1	0	0	0	0
plasm1	0.45	1		0.63	0.59	0.4	0.05	0.06	0	0	0	0
cross3	0.96		1	0.1	0.08	0.04	0	0	0	0	0	0
sms		0.96	0.45	0	0	0	0	0	0	0	0	0

[illegible]

cross0	0	0	0	0	0	0	0	0	0	0.2	0.84	0.87	
plasm0	0	0	0	0	0	0	0	0	0.15	1	1		0.87
cross1	0	0	0	0	0	0	0	0	0.17	1		1	0.84
plasm3	0	0	0	0	0	0	0	0.1	0.81		1	1	0.2
moead	0	0	0	0.01	0.21	0.45	0.52	0.99		0.81	0.17	0.15	0
gde	0	0	0	0.25	0.94	0.99	1		0.99	0.1	0	0	0
plasm2	0	0.03	0.05	0.93	1	1		1	0.52	0	0	0	0
ssnsga	0	0.04	0.06	0.96	1		1	0.99	0.45	0	0	0	0
cross2	0	0.12	0.18	1		1	1	0.94	0.21	0	0	0	0
nsga	0.01	0.8	0.88		1	0.96	0.93	0.25	0.01	0	0	0	0
cross3	0.71	1		0.88	0.18	0.06	0.05	0	0	0	0	0	0
plasm1	0.81		1	0.8	0.12	0.04	0.03	0	0	0	0	0	0
sms		0.81	0.71	0.01	0	0	0	0	0	0	0	0	0
	sms	plasm1	cross3	nsga	cross2	ssnsga	plasm2	gde	moead	plasm3	cross1	plasm0	cross0

	plasm3	cross3	cross0	nsga	moead	cross2	cross1	sms	gde	ssnsga	plasm0	plasm1	plasm2
plasm2	0	0	0	0	0	0	0	0	0.02	0.02	0.69	1	
plasm1	0	0	0	0	0	0	0	0	0.32	0.33	1		1
plasm0	0	0	0	0	0	0	0	0.1	0.95	0.95		1	0.69
ssnsga	0	0	0	0	0.02	0.32	0.95	1		0.95	0.33	0.02	
gde	0	0	0	0	0.02	0.33	0.95		1	0.95	0.32	0.02	
sms	0	0	0	0.02	0.02	0.69	1		0.95	0.95	0.1	0	0
cross1	0	0	0	0.31	0.33	1		1	0.33	0.32	0	0	0
cross2	0	0	0.08	0.94	0.95		1	0.69	0.02	0.02	0	0	0
moead	0.02	0.35	0.92	1		0.95	0.33	0.02	0	0	0	0	0
nsga	0.02	0.38	0.93		1	0.94	0.31	0.02	0	0	0	0	0
cross0	0.73	1		0.93	0.92	0.08	0	0	0	0	0	0	0
cross3	1		1	0.38	0.35	0	0	0	0	0	0	0	0
plasm3		1	0.73	0.02	0.02	0	0	0	0	0	0	0	0

Figura 122 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_14 e $W = 15$.

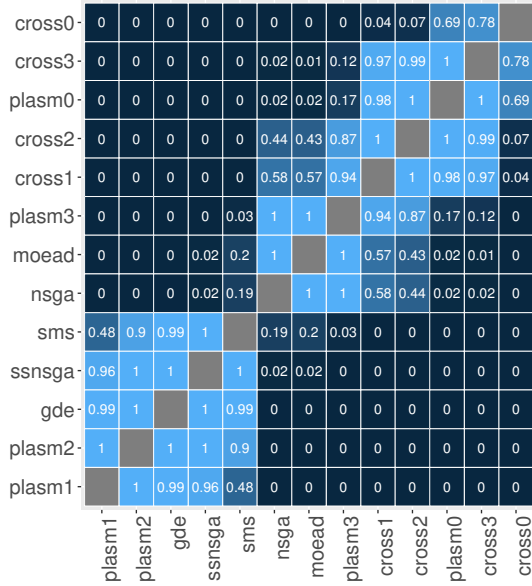


Figura 124 – *Posthoc* de Nemenyi I_H para b30_14 e $W = 15$.

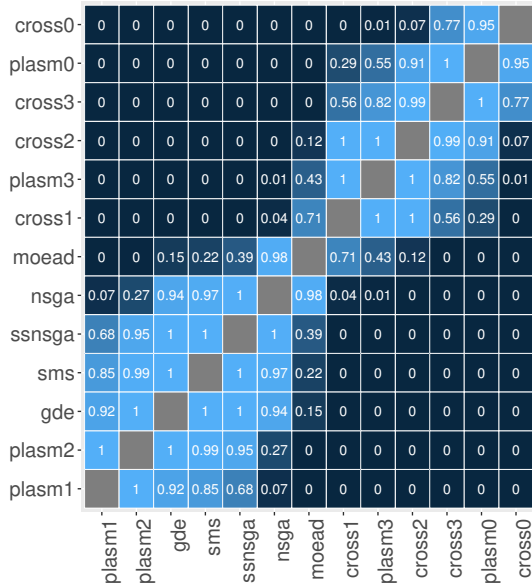


Figura 123 – *Posthoc* de Nemenyi I_{ϵ} para b30_14 e $W = 15$.

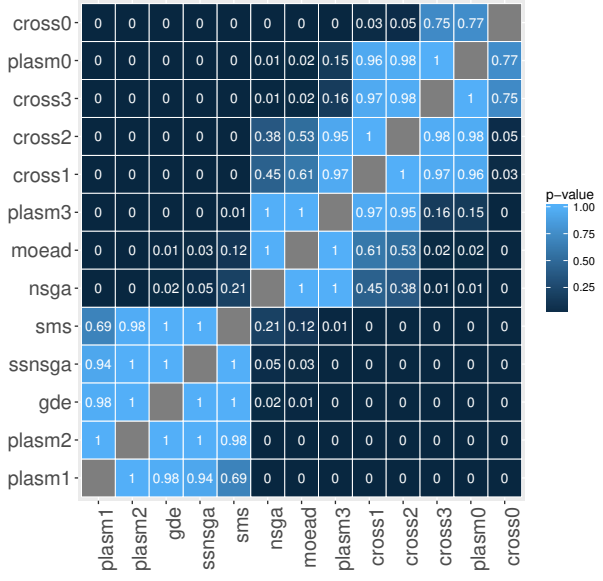


Figura 125 – *Posthoc* de Nemenyi Tempo(s) para b30_14 e $W = 15$.

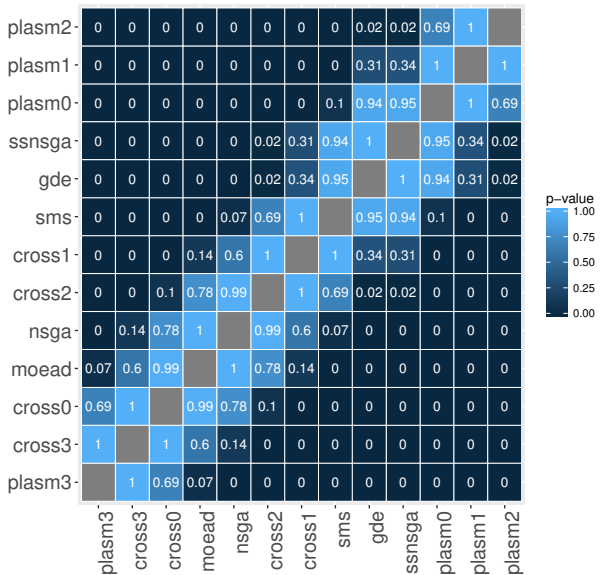


Figura 126 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_15 e $W = 15$.

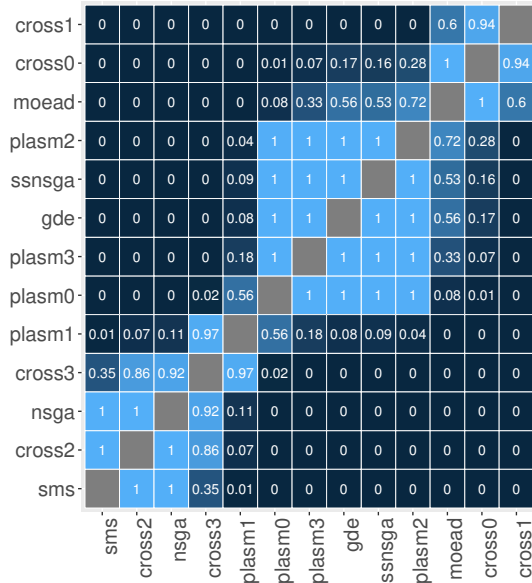


Figura 128 – *Posthoc* de Nemenyi I_H para b30_15 e $W = 15$.

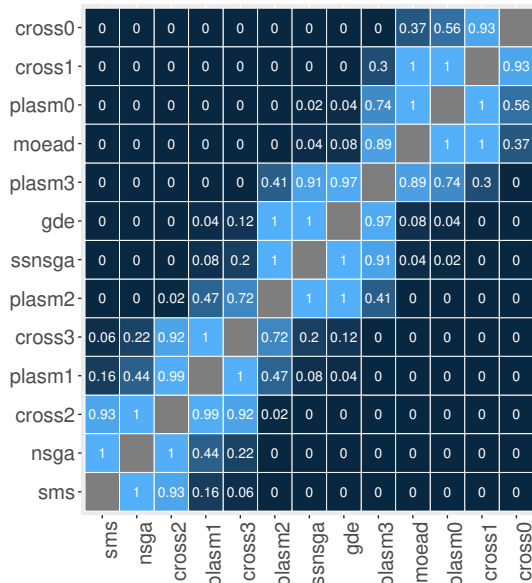


Figura 127 – *Posthoc* de Nemenyi I_{ϵ} para b30_15 e $W = 15$.

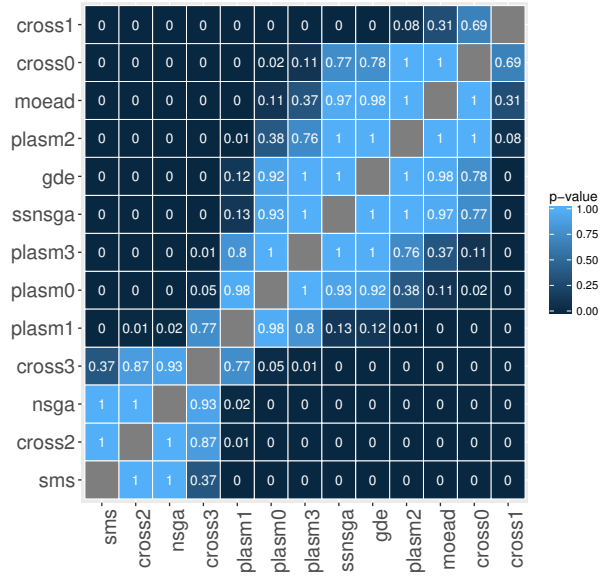


Figura 129 – *Posthoc* de Nemenyi Tempo(s) para b30_15 e $W = 15$.

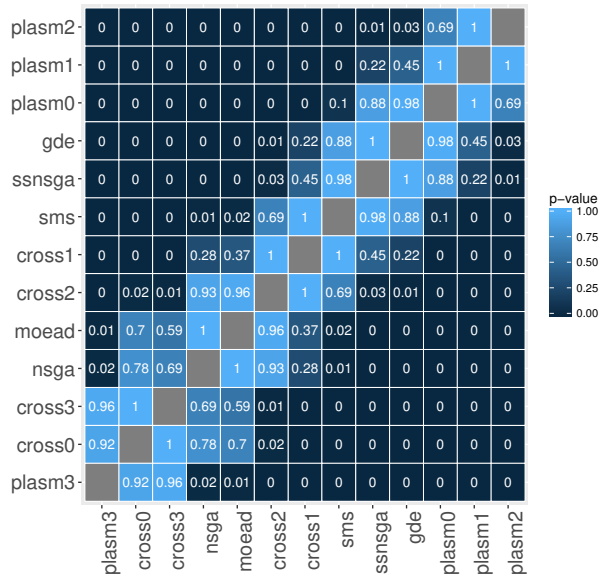


Figura 130 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_19 e $W = 20$.

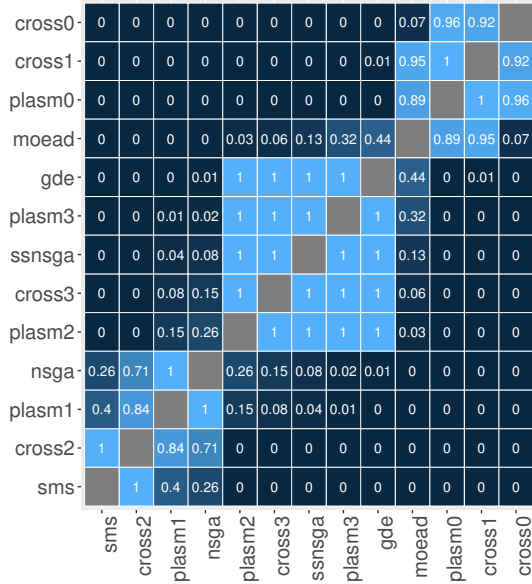


Figura 131 – *Posthoc* de Nemenyi I_{ϵ} para b30_19 e $W = 20$.

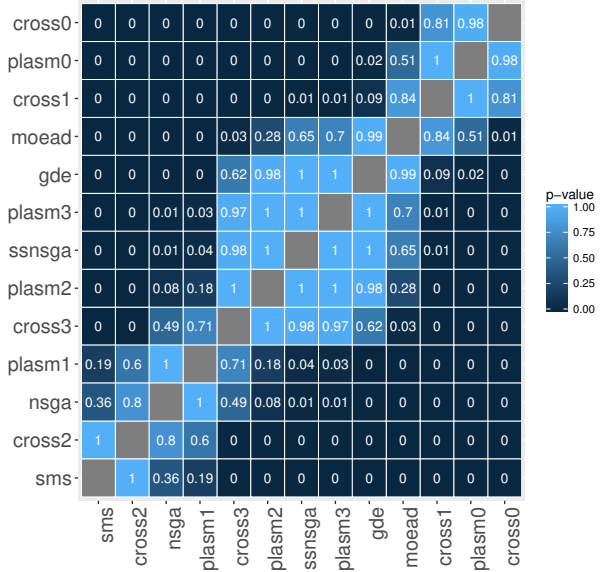


Figura 132 – *Posthoc* de Nemenyi I_H para b30_19 e $W = 20$.

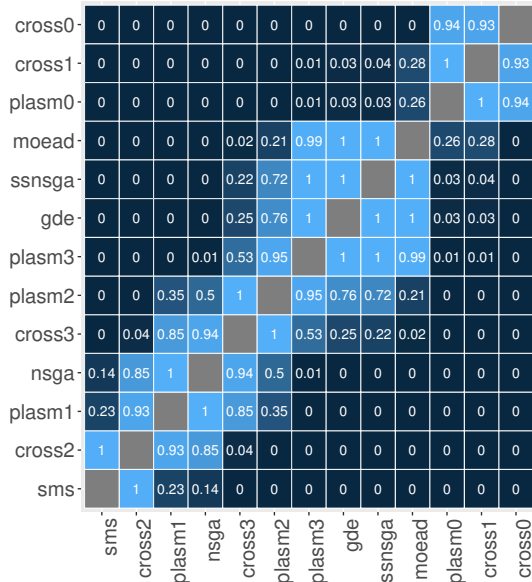


Figura 133 – *Posthoc* de Nemenyi Tempo(s) para b30_19 e $W = 20$.

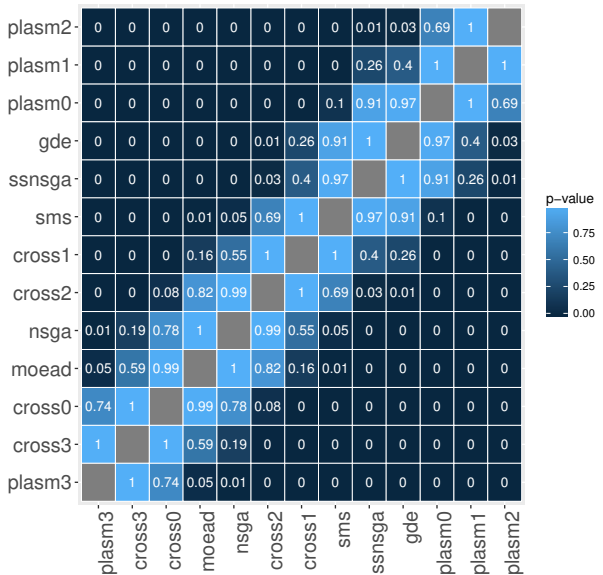


Figura 134 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_20 e $W = 20$.

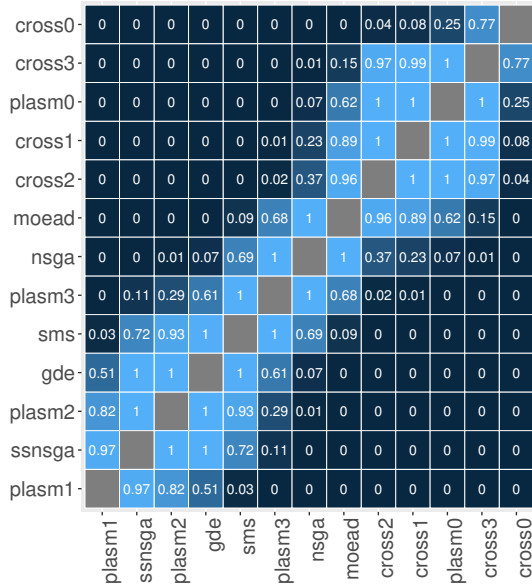


Figura 136 – *Posthoc* de Nemenyi I_H para b30_20 e $W = 20$.

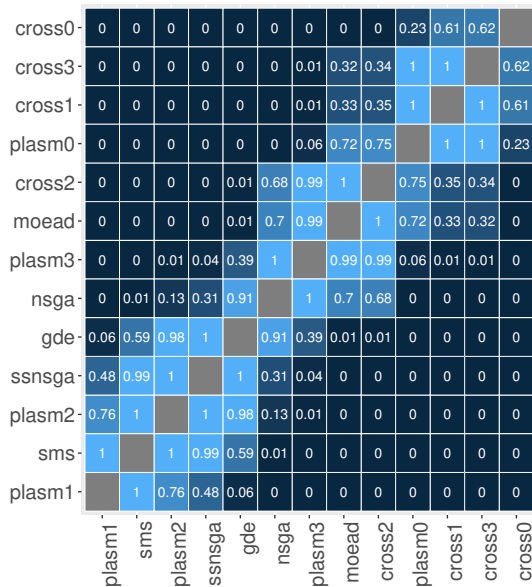


Figura 135 – *Posthoc* de Nemenyi I_{ϵ} para b30_20 e $W = 20$.

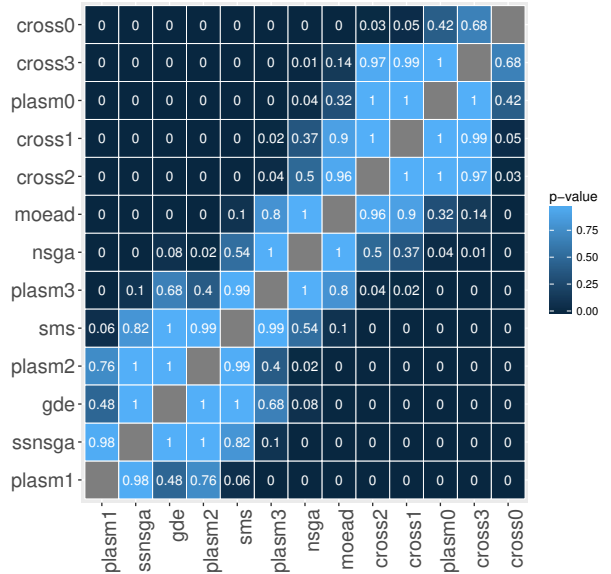


Figura 137 – *Posthoc* de Nemenyi Tempo(s) para b30_20 e $W = 20$.

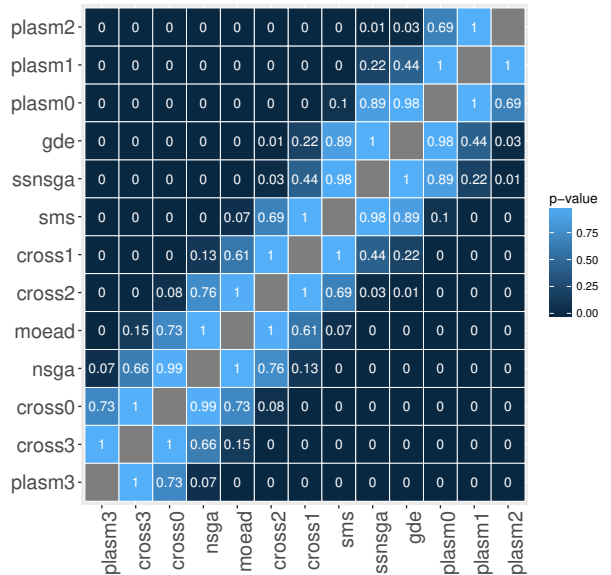


Figura 138 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_21 e $W = 20$.

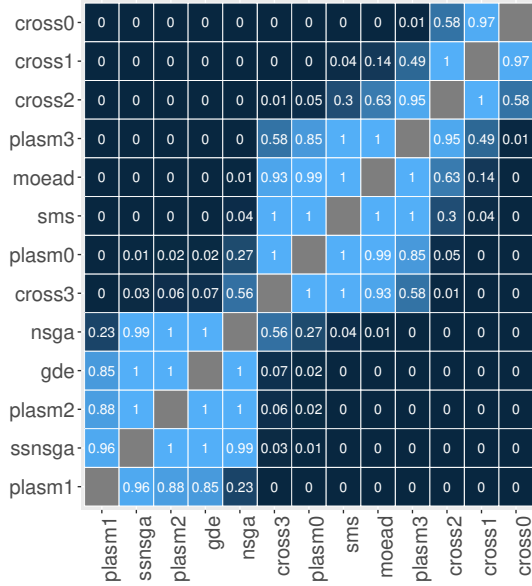


Figura 140 – *Posthoc* de Nemenyi I_H para b30_21 e $W = 20$.

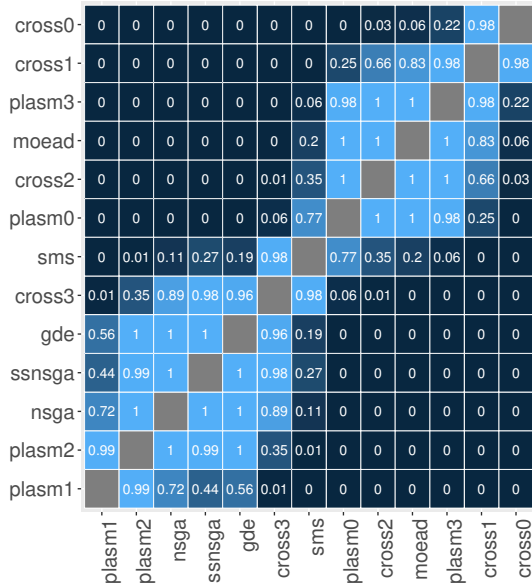


Figura 139 – *Posthoc* de Nemenyi I_{ϵ} para b30_21 e $W = 20$.

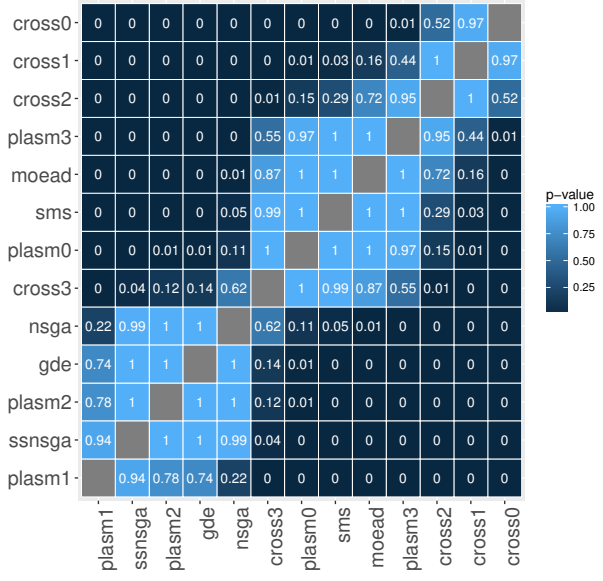


Figura 141 – *Posthoc* de Nemenyi Tempo(s) para b30_21 e $W = 20$.

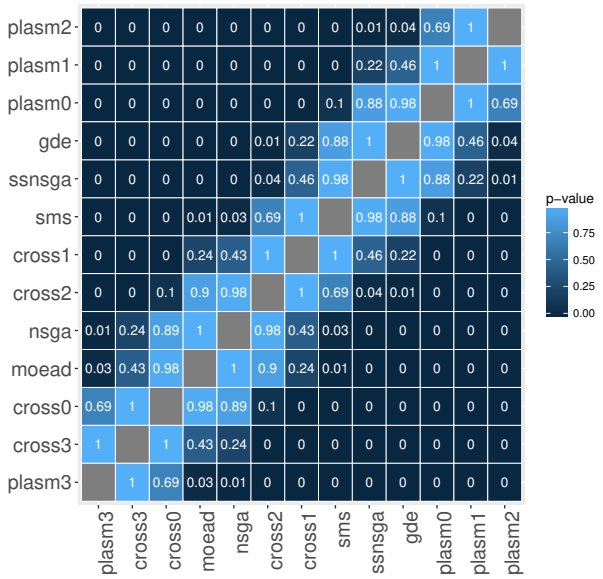


Figura 142 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_25 e $W = 25$.

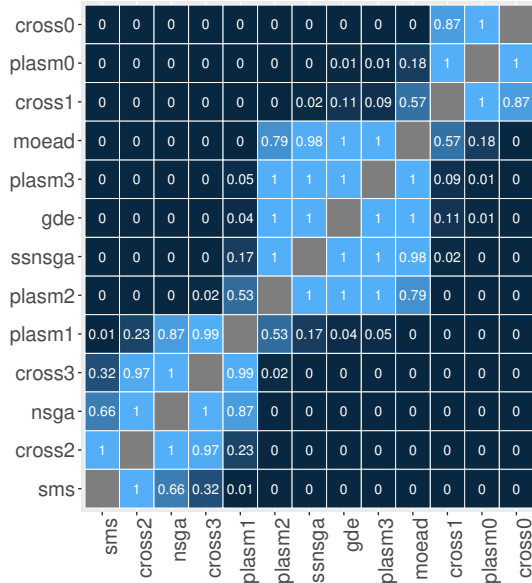


Figura 144 – *Posthoc* de Nemenyi I_H para b30_25 e $W = 25$.

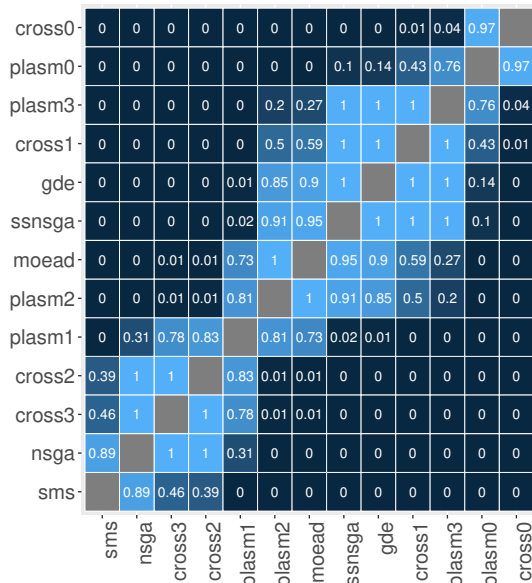


Figura 143 – *Posthoc* de Nemenyi I_{ϵ} para b30_25 e $W = 25$.

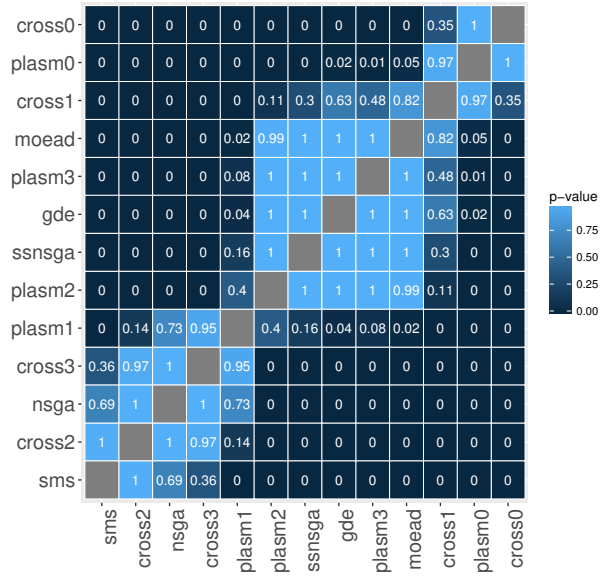
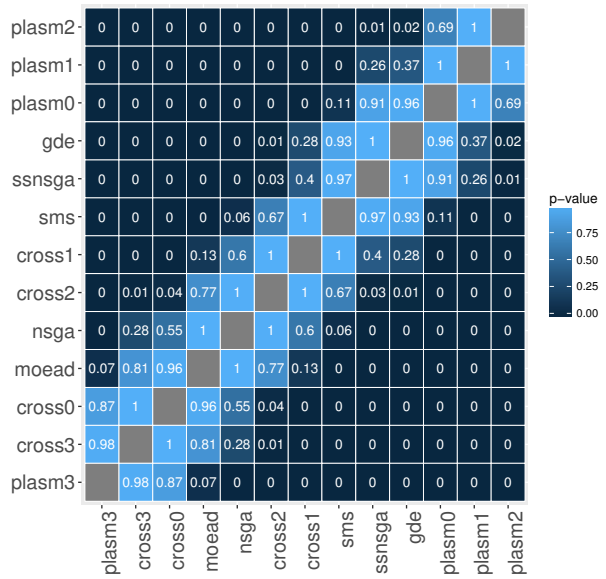


Figura 145 – *Posthoc* de Nemenyi Tempo(s) para b30_25 e $W = 25$.



D. RESULTADOS DO MODEO MULTIOBJETIVO EXPERIMENTO 2

Figura 146 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_26 e $W = 25$.

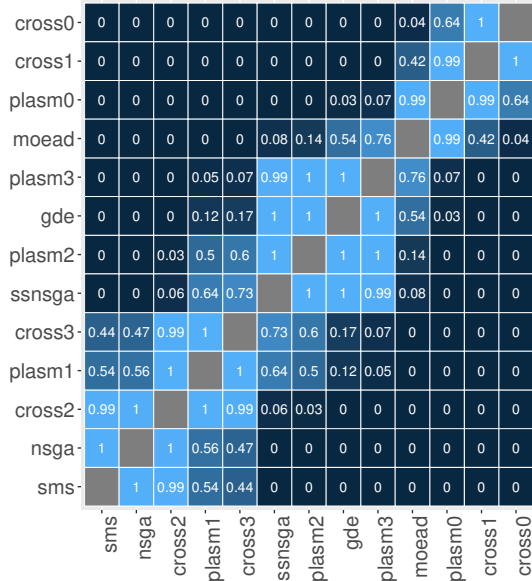


Figura 148 – *Posthoc* de Nemenyi I_H para b30_26 e $W = 25$.

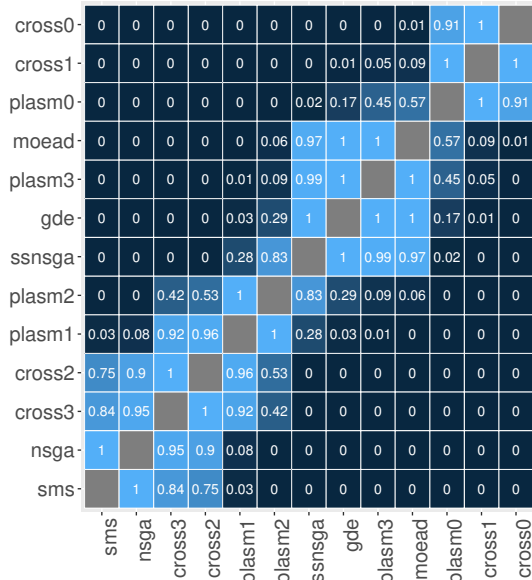


Figura 147 – *Posthoc* de Nemenyi I_{ϵ} para b30_26 e $W = 25$.

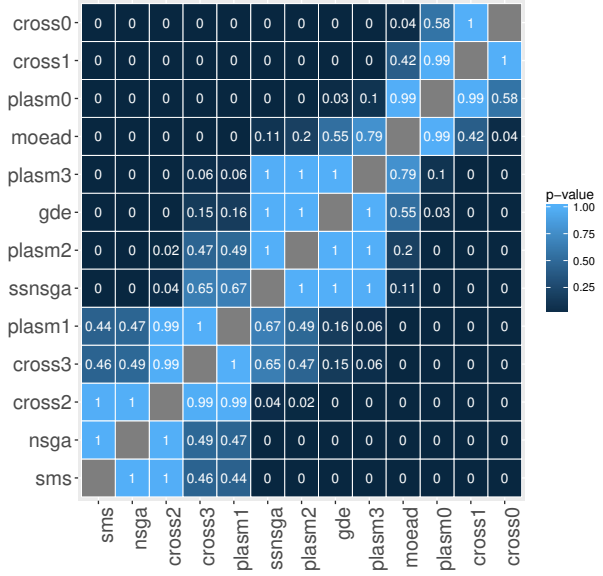


Figura 149 – *Posthoc* de Nemenyi Tempo(s) para b30_26 e $W = 25$.

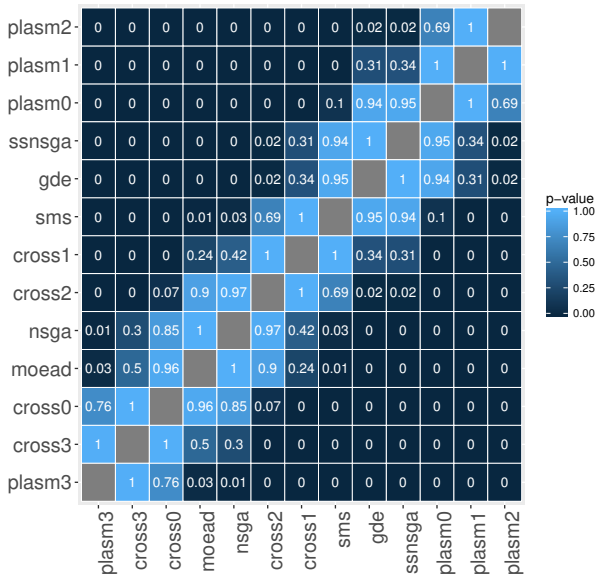


Figura 150 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b30_27 e $W = 25$.

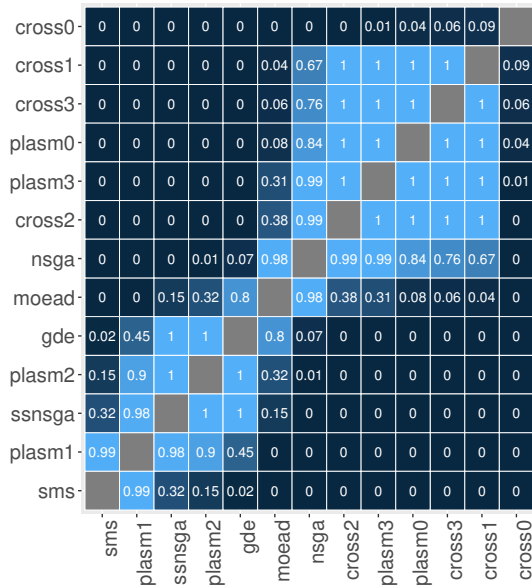


Figura 151 – *Posthoc* de Nemenyi I_{ϵ} para b30_27 e $W = 25$.

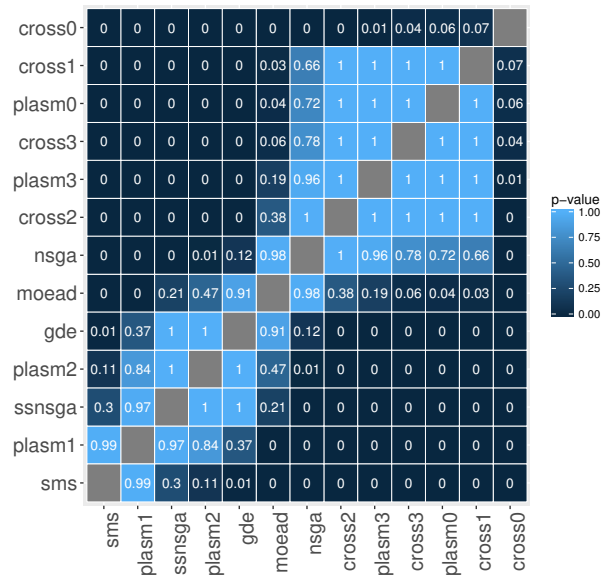


Figura 152 – *Posthoc* de Nemenyi I_H para b30_27 e $W = 25$.

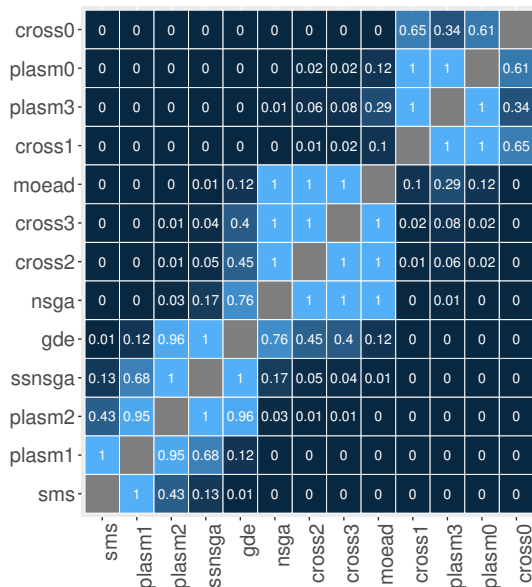
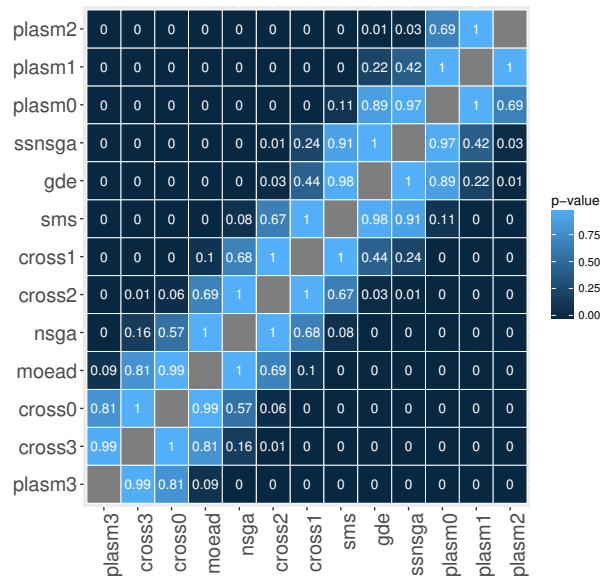


Figura 153 – *Posthoc* de Nemenyi Tempo(s) para b30_27 e $W = 25$.



D. RESULTADOS DO MODEO MULTIOBJETIVO EXPERIMENTO 2

Figura 154 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_01 e $W = 5$.

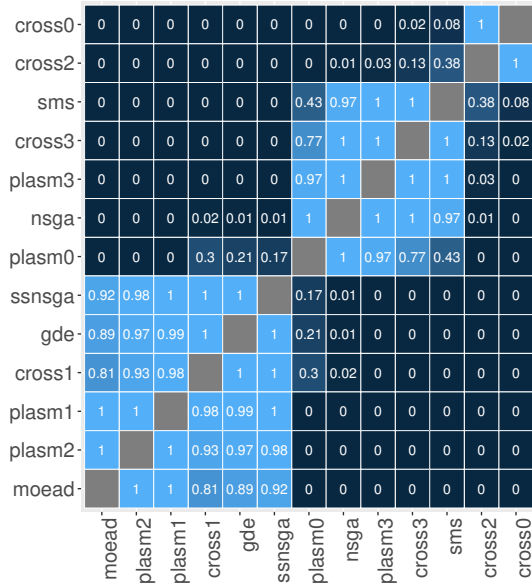


Figura 156 – *Posthoc* de Nemenyi I_H para b60_01 e $W = 5$.

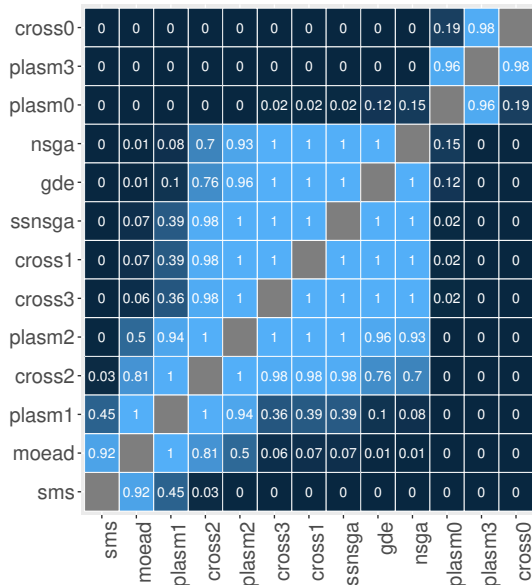


Figura 155 – *Posthoc* de Nemenyi I_{ϵ} para b60_01 e $W = 5$.

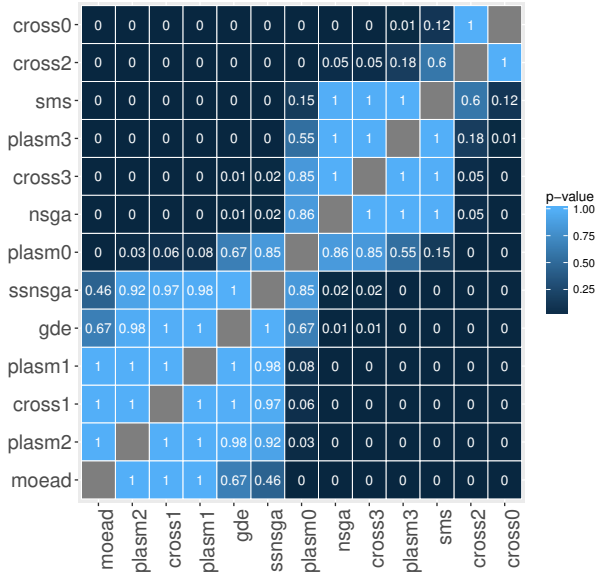


Figura 157 – *Posthoc* de Nemenyi Tempo(s) para b60_01 e $W = 5$.

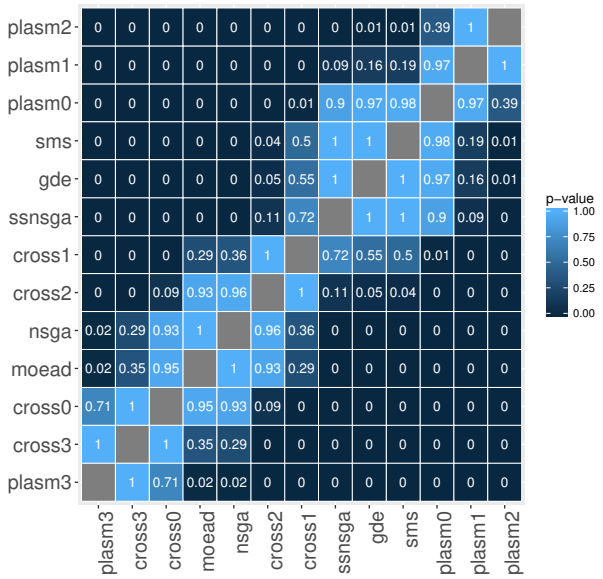


Figura 158 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_02 e $W = 5$.

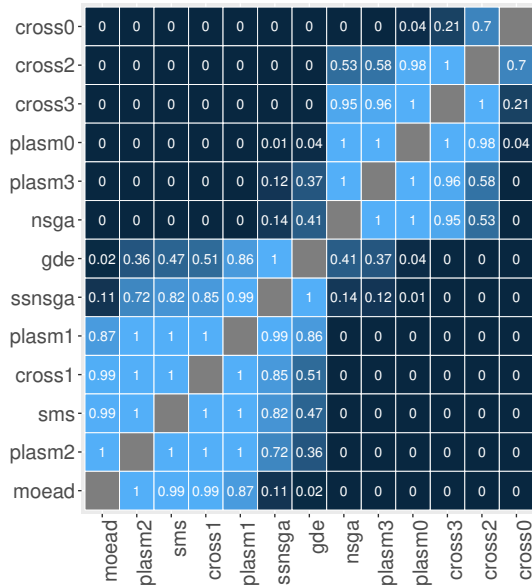


Figura 160 – *Posthoc* de Nemenyi I_H para b60_02 e $W = 5$.

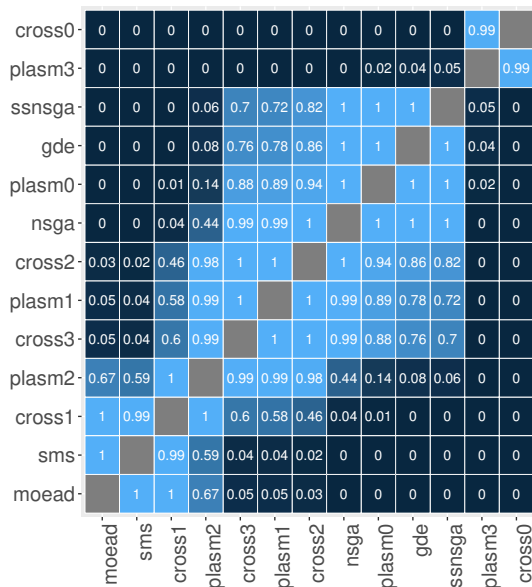


Figura 159 – *Posthoc* de Nemenyi I_{ϵ} para b60_02 e $W = 5$.

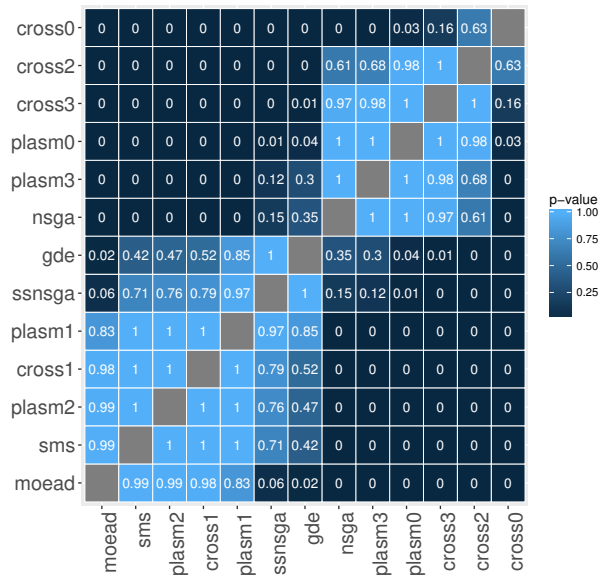


Figura 161 – *Posthoc* de Nemenyi Tempo(s) para b60_02 e $W = 5$.

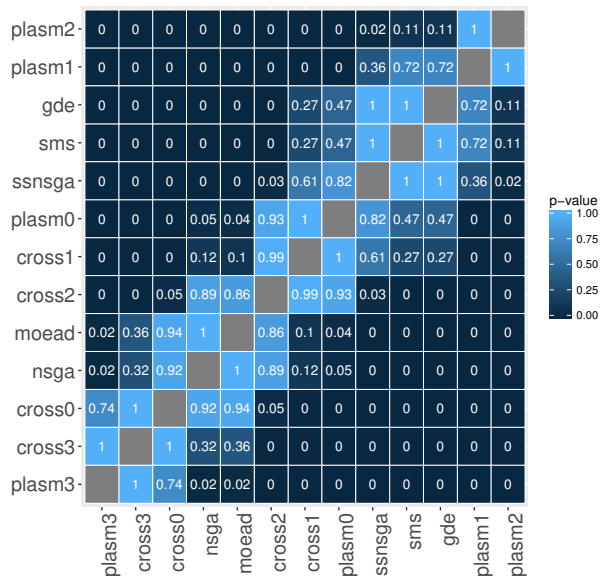


Figura 162 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_03 e $W = 5$.

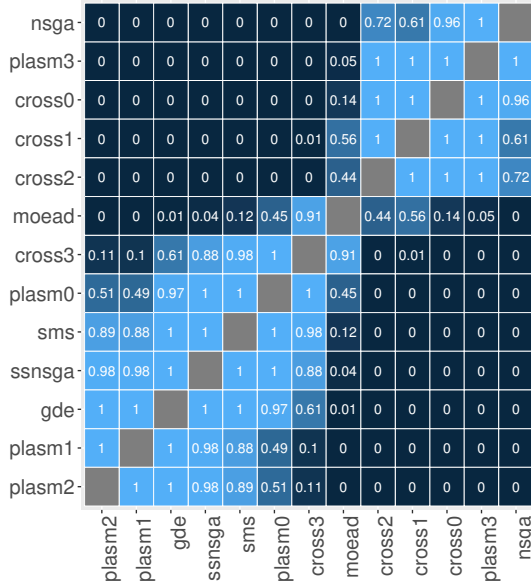


Figura 164 – *Posthoc* de Nemenyi I_H para b60_03 e $W = 5$.

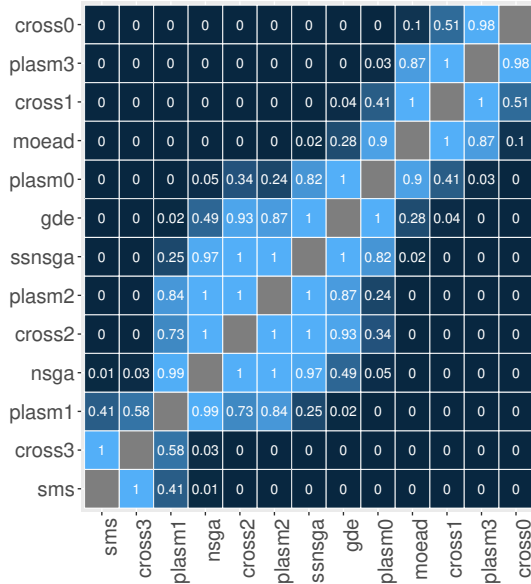


Figura 163 – *Posthoc* de Nemenyi I_{ϵ} para b60_03 e $W = 5$.

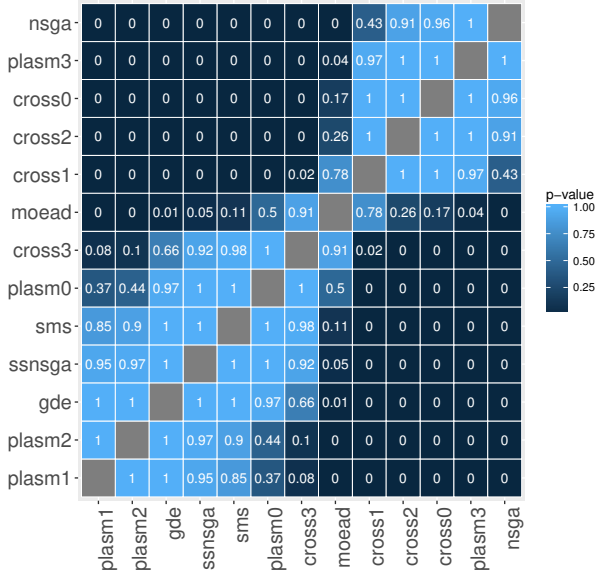


Figura 165 – *Posthoc* de Nemenyi Tempo(s) para b60_03 e $W = 5$.

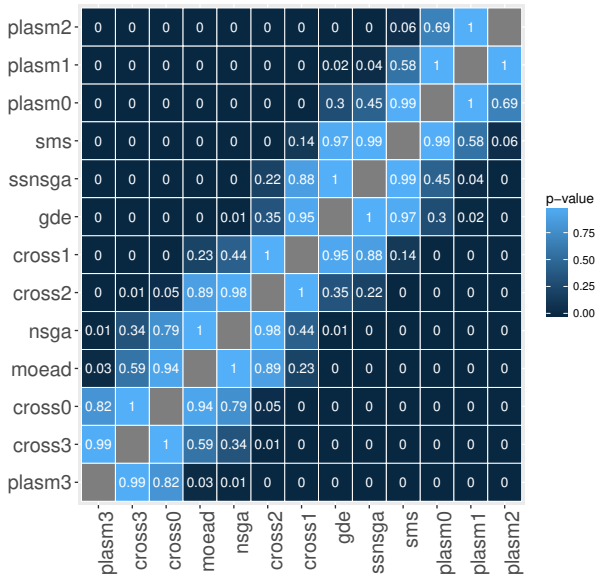


Figura 166 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_07 e $W = 10$.

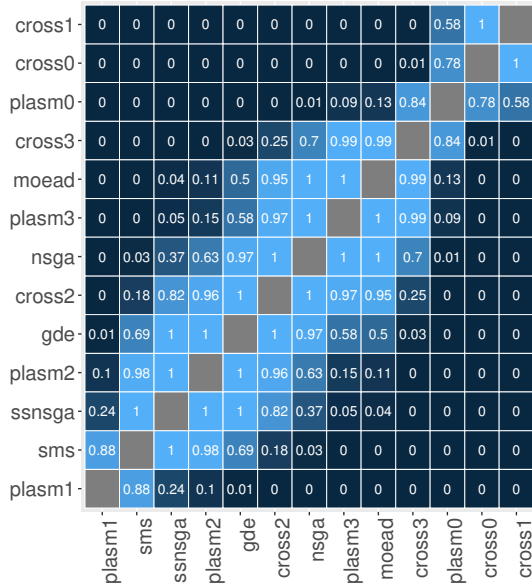


Figura 168 – *Posthoc* de Nemenyi I_H para b60_07 e $W = 10$.

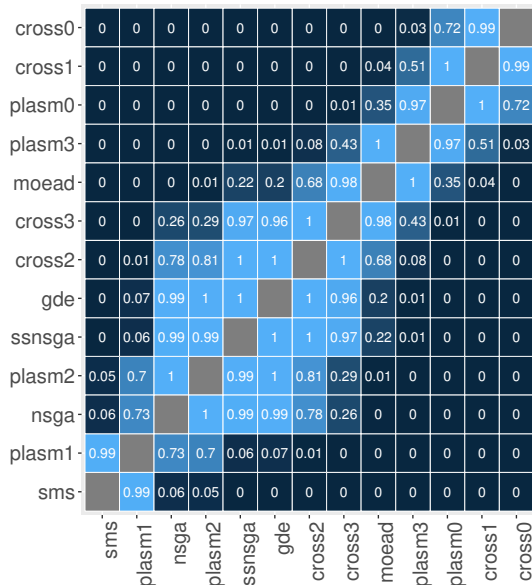


Figura 167 – *Posthoc* de Nemenyi I_{ϵ} para b60_07 e $W = 10$.

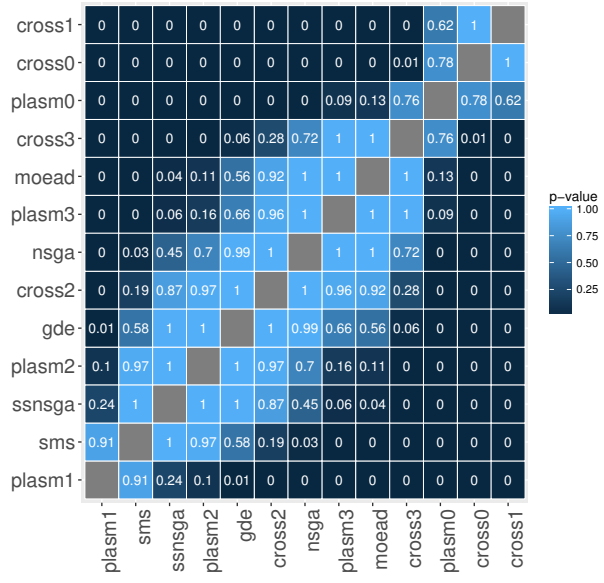


Figura 169 – *Posthoc* de Nemenyi Tempo(s) para b60_07 e $W = 10$.

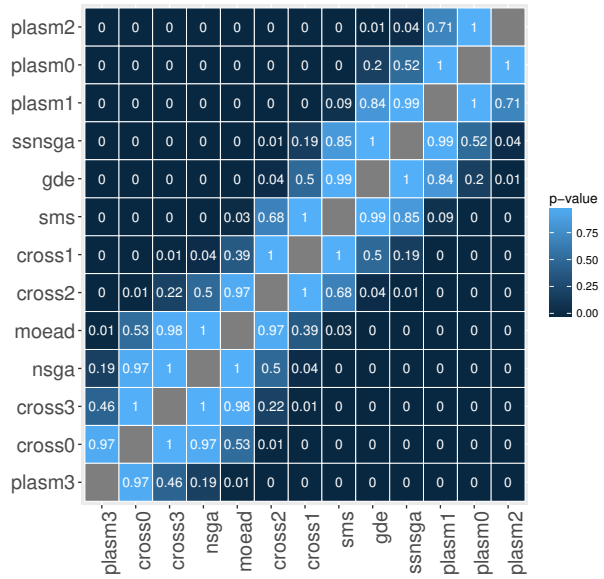


Figura 170 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_08 e $W = 10$.

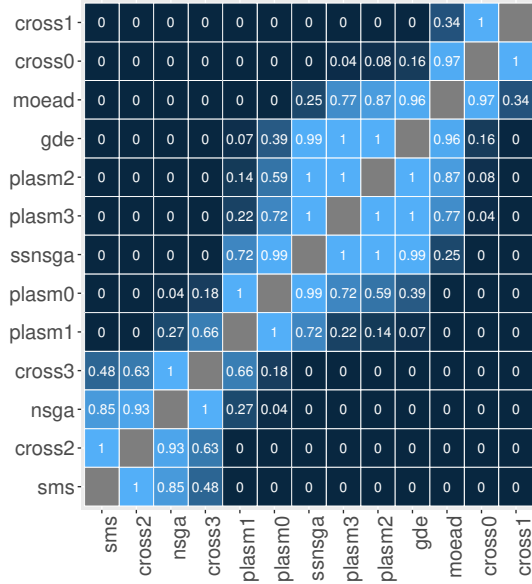


Figura 172 – *Posthoc* de Nemenyi I_H para b60_08 e $W = 10$.

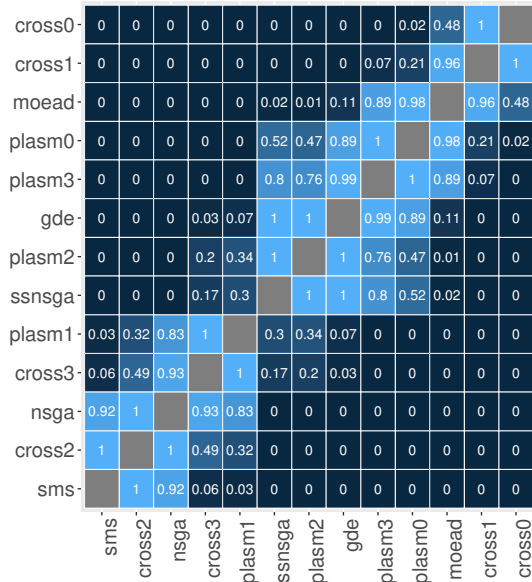


Figura 171 – *Posthoc* de Nemenyi I_{ϵ} para b60_08 e $W = 10$.

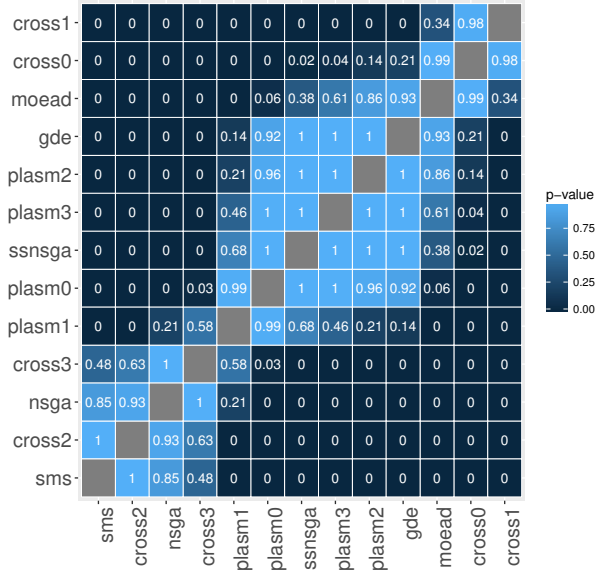


Figura 173 – *Posthoc* de Nemenyi Tempo(s) para b60_08 e $W = 10$.

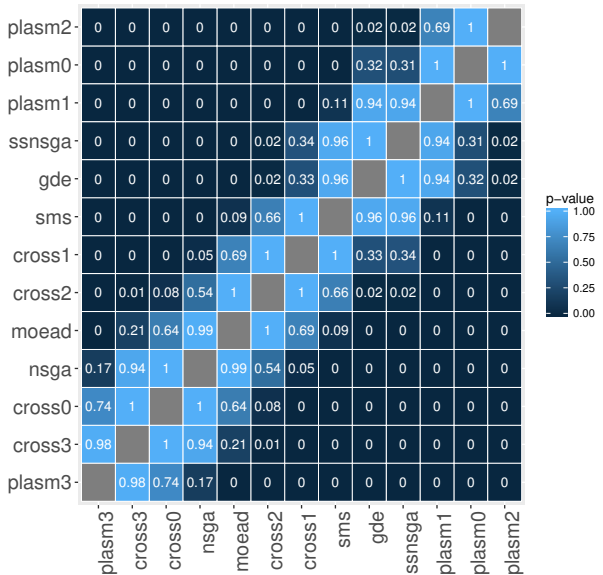


Figura 174 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_09 e $W = 10$.

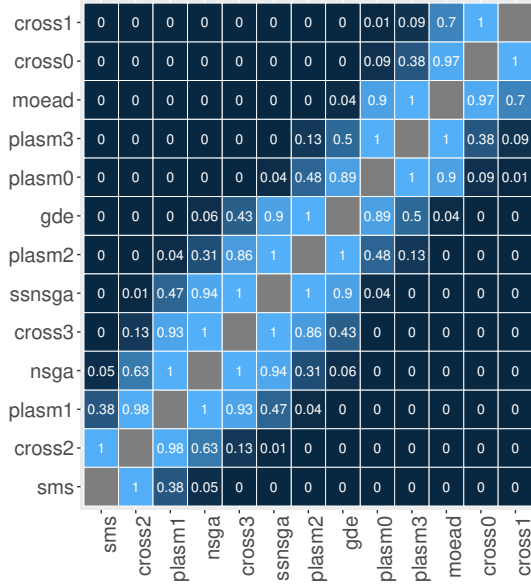


Figura 176 – *Posthoc* de Nemenyi I_H para b60_09 e $W = 10$.

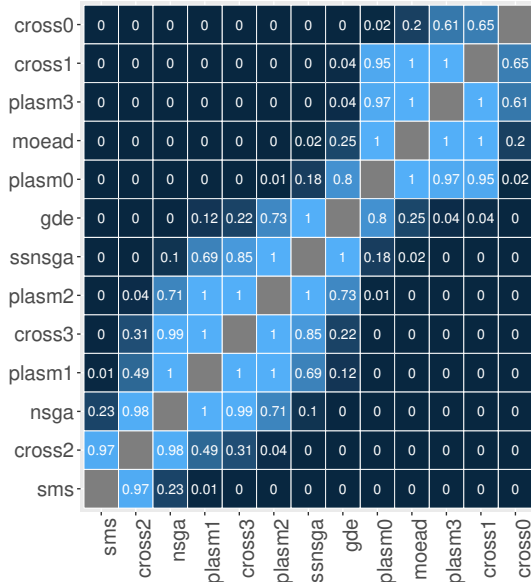


Figura 175 – *Posthoc* de Nemenyi I_{ϵ} para b60_09 e $W = 10$.

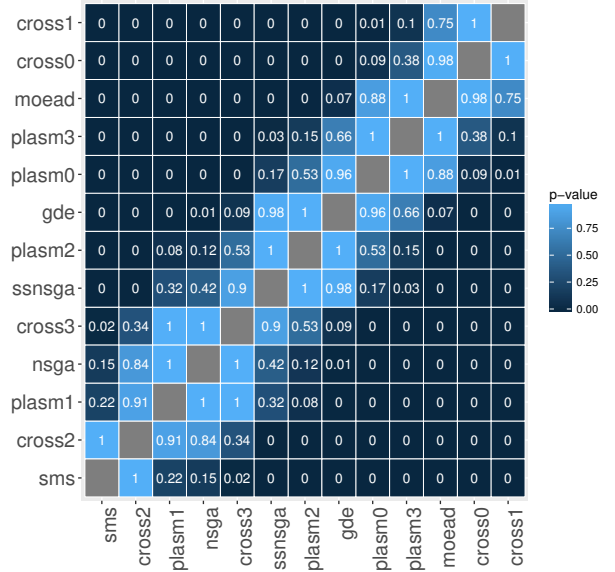


Figura 177 – *Posthoc* de Nemenyi Tempo(s) para b60_09 e $W = 10$.

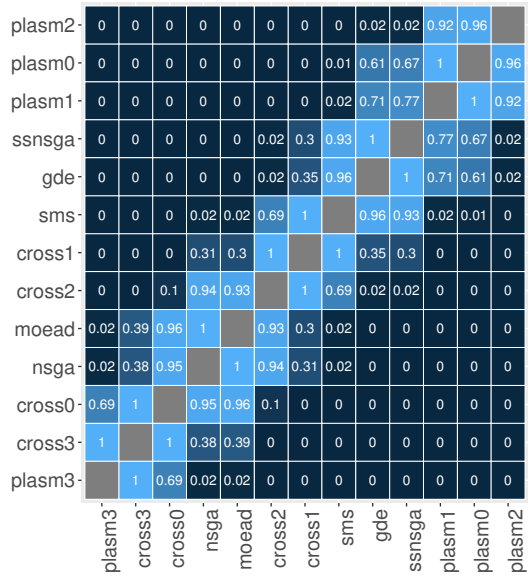


Figura 178 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_13 e $W = 15$.

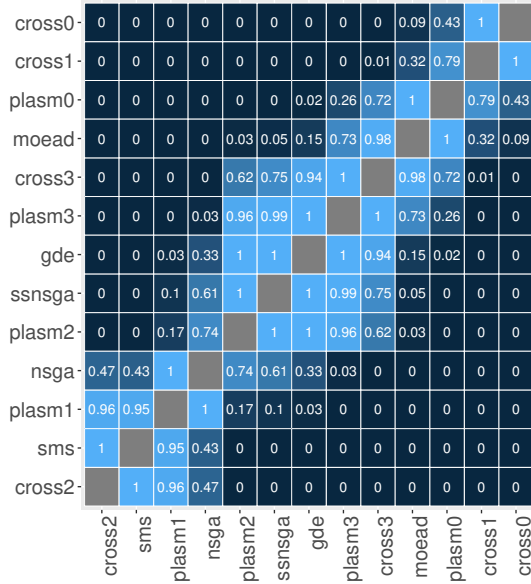


Figura 180 – *Posthoc* de Nemenyi I_H para b60_13 e $W = 15$.

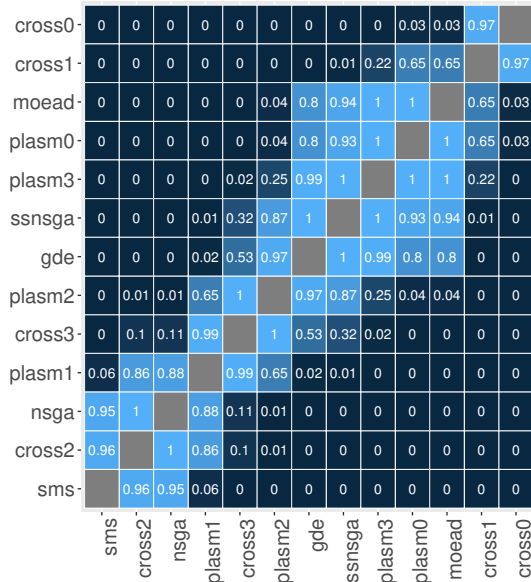


Figura 179 – *Posthoc* de Nemenyi I_{ϵ} para b60_13 e $W = 15$.

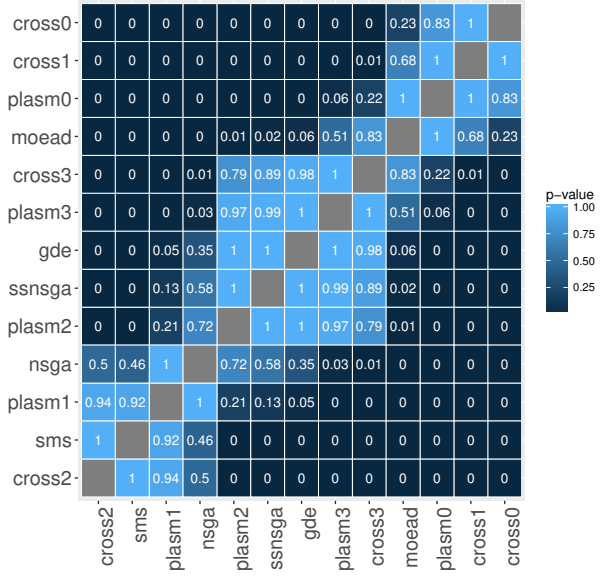


Figura 181 – *Posthoc* de Nemenyi Tempo(s) para b60_13 e $W = 15$.

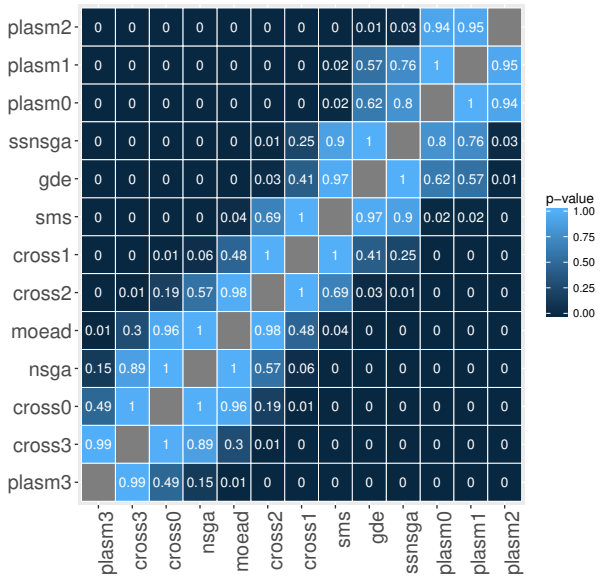


Figura 182 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_14 e $W = 15$.

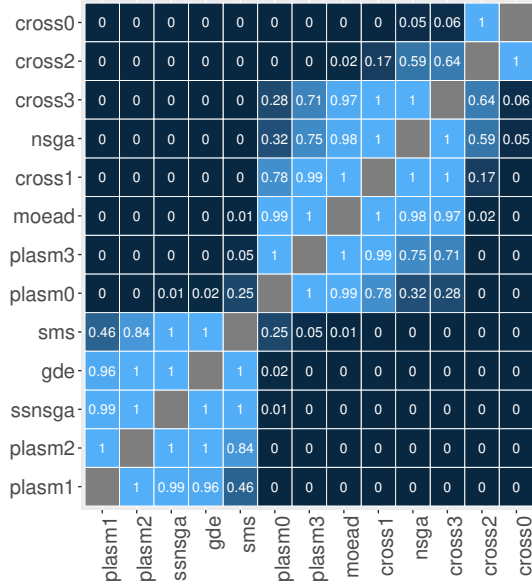


Figura 184 – *Posthoc* de Nemenyi I_H para b60_14 e $W = 15$.

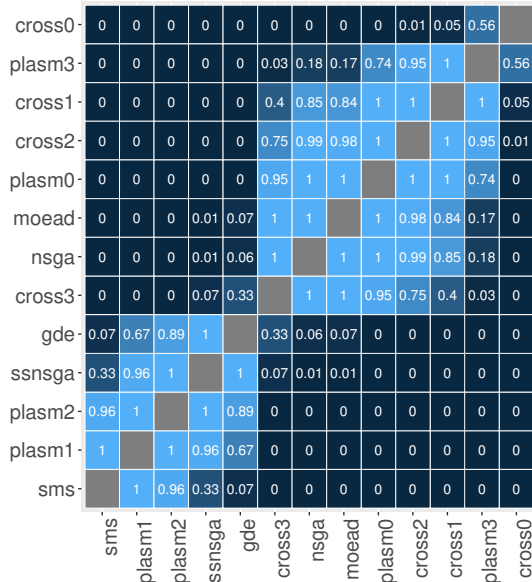


Figura 183 – *Posthoc* de Nemenyi I_{ϵ} para b60_14 e $W = 15$.

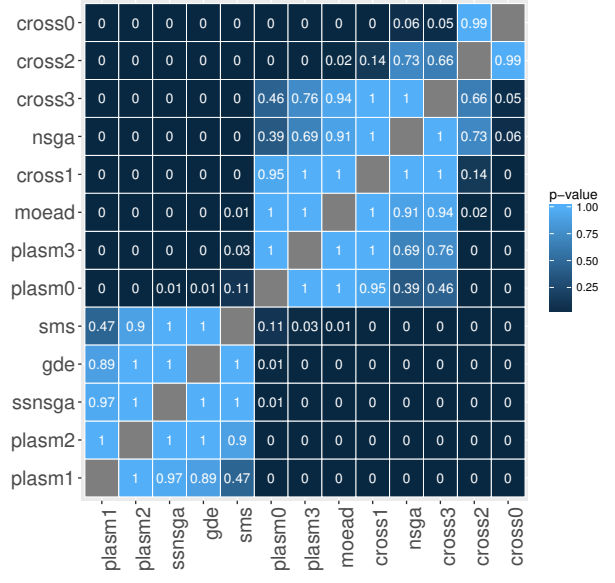
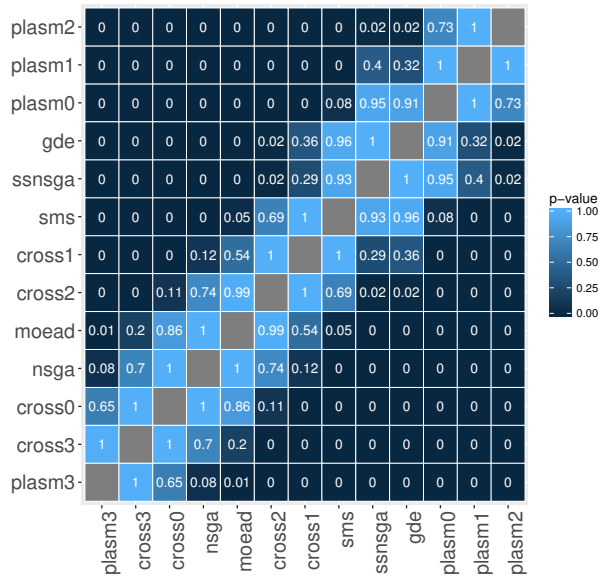


Figura 185 – *Posthoc* de Nemenyi Tempo(s) para b60_14 e $W = 15$.



D. RESULTADOS DO MODEO MULTIOBJETIVO EXPERIMENTO 2

Figura 186 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_15 e $W = 15$.

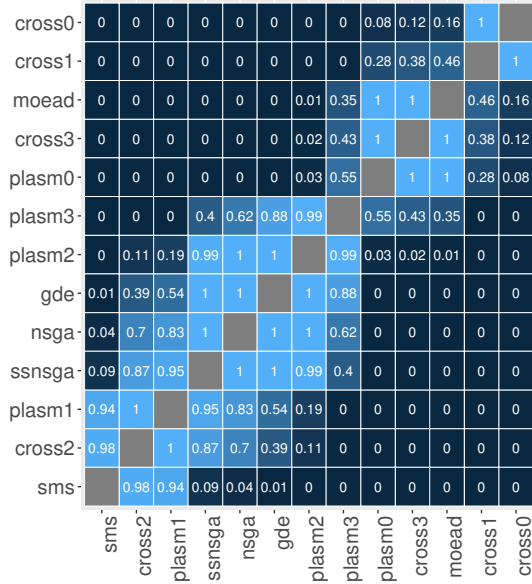


Figura 188 – *Posthoc* de Nemenyi I_H para b60_15 e $W = 15$.

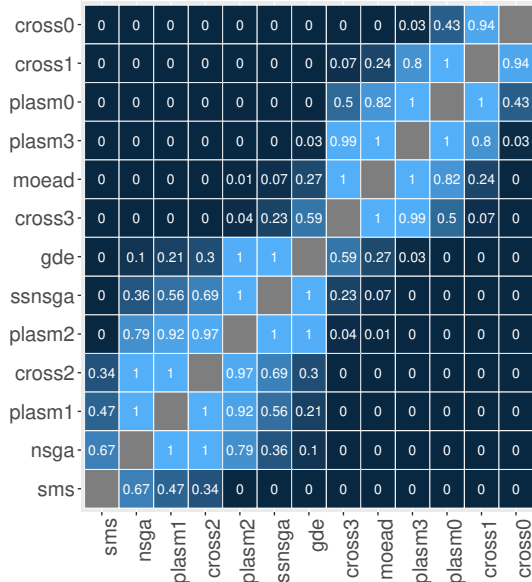


Figura 187 – *Posthoc* de Nemenyi I_{ϵ} para b60_15 e $W = 15$.

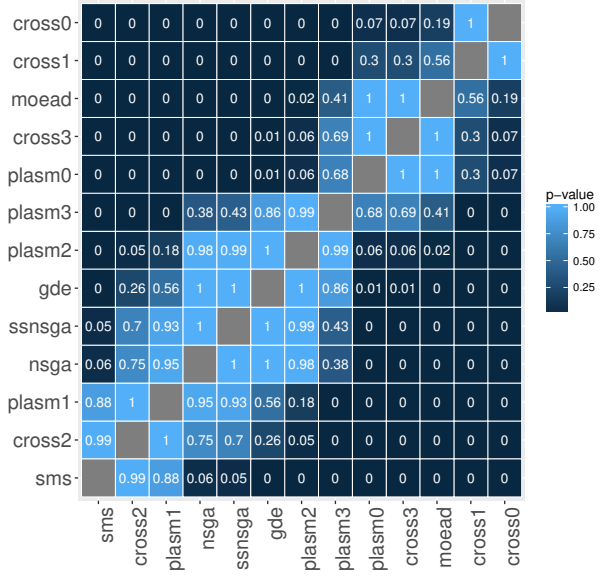


Figura 189 – *Posthoc* de Nemenyi Tempo(s) para b60_15 e $W = 15$.

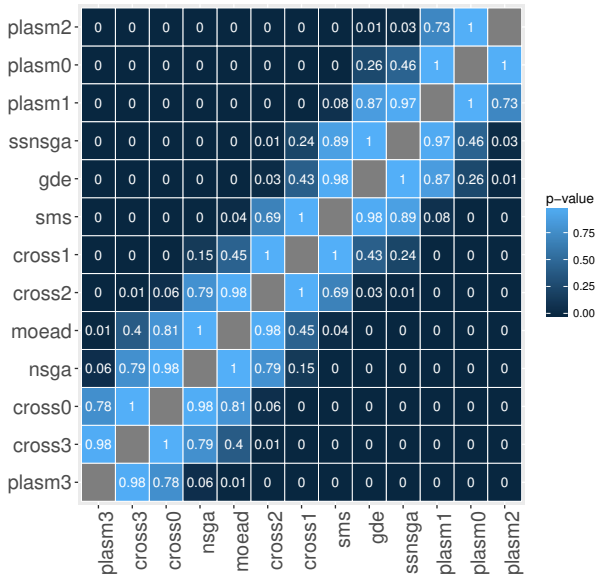


Figura 190 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_19 e $W = 20$.

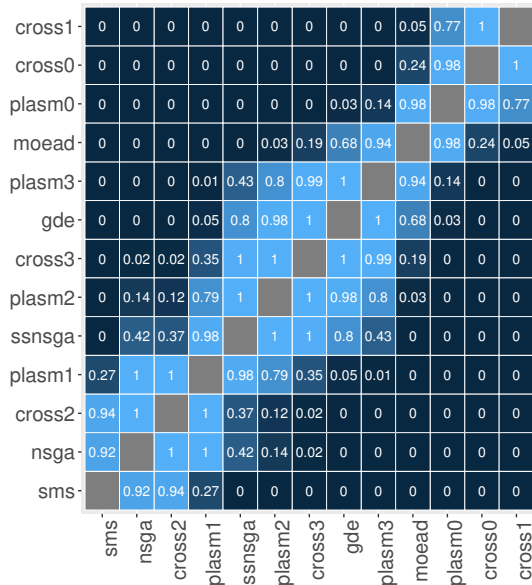


Figura 191 – *Posthoc* de Nemenyi I_{ϵ} para b60_19 e $W = 20$.

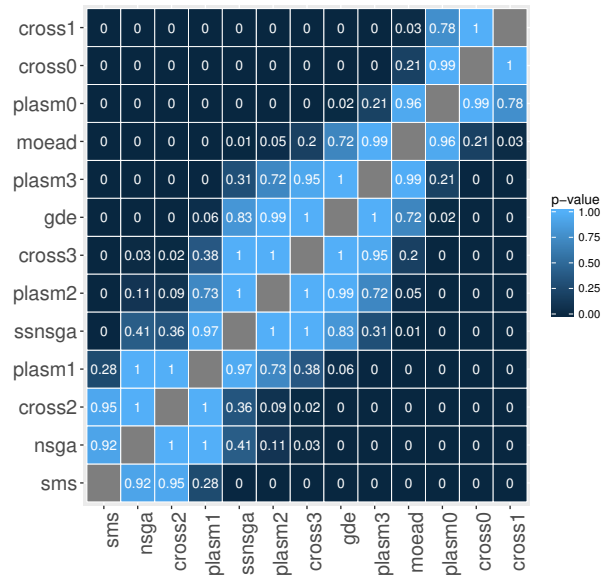


Figura 192 – *Posthoc* de Nemenyi I_H para b60_19 e $W = 20$.

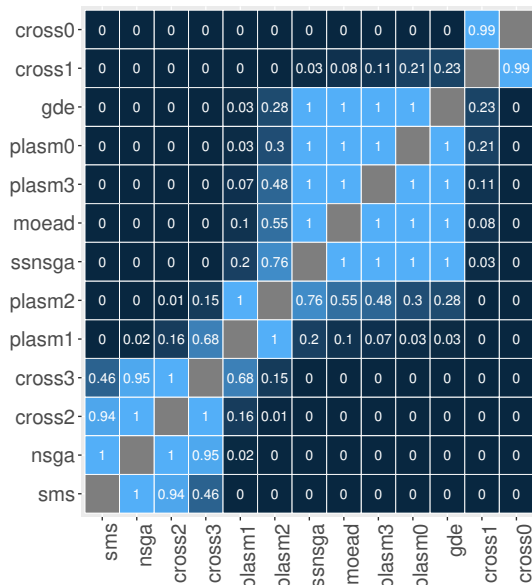


Figura 193 – *Posthoc* de Nemenyi Tempo(s) para b60_19 e $W = 20$.

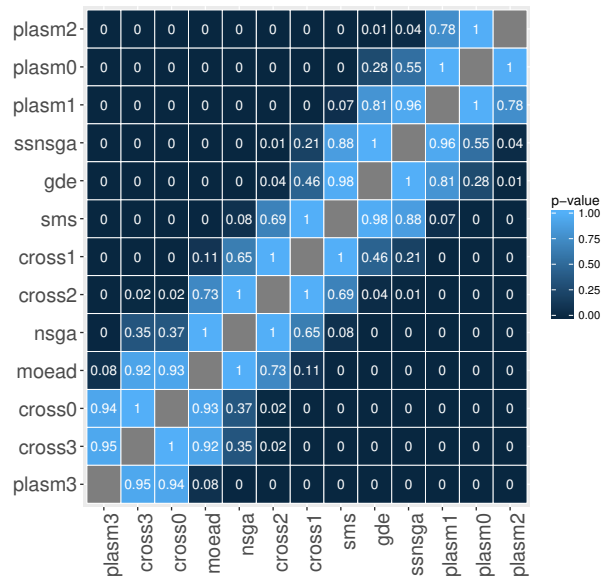


Figura 194 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_20 e $W = 20$.

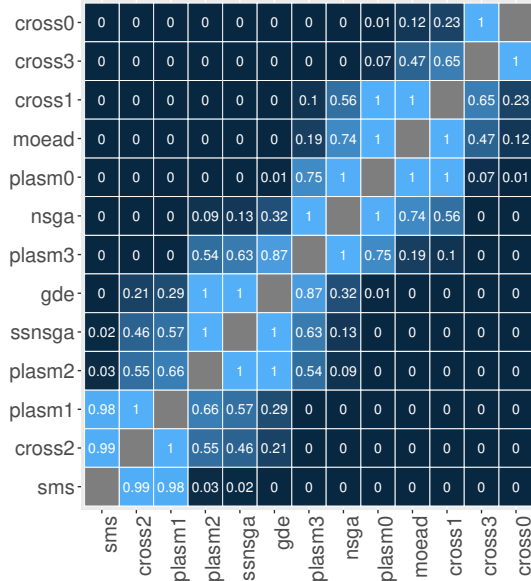


Figura 196 – *Posthoc* de Nemenyi I_H para b60_20 e $W = 20$.

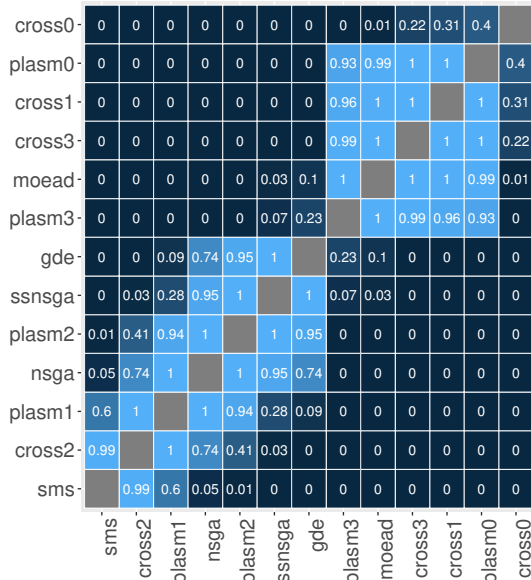


Figura 195 – *Posthoc* de Nemenyi I_{ϵ} para b60_20 e $W = 20$.

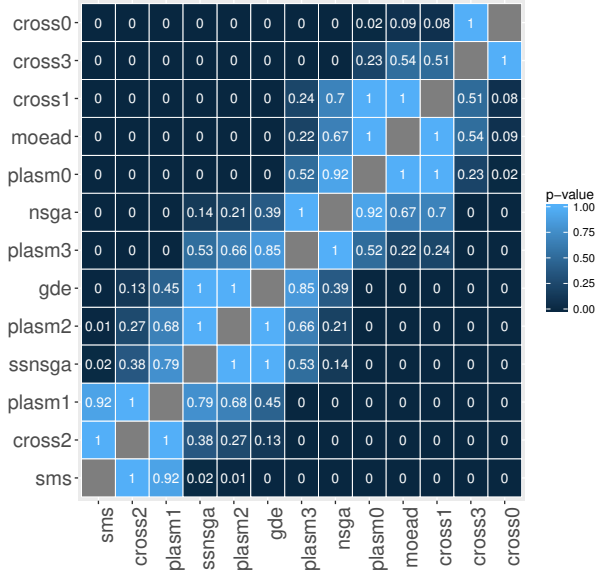


Figura 197 – *Posthoc* de Nemenyi Tempo(s) para b60_20 e $W = 20$.

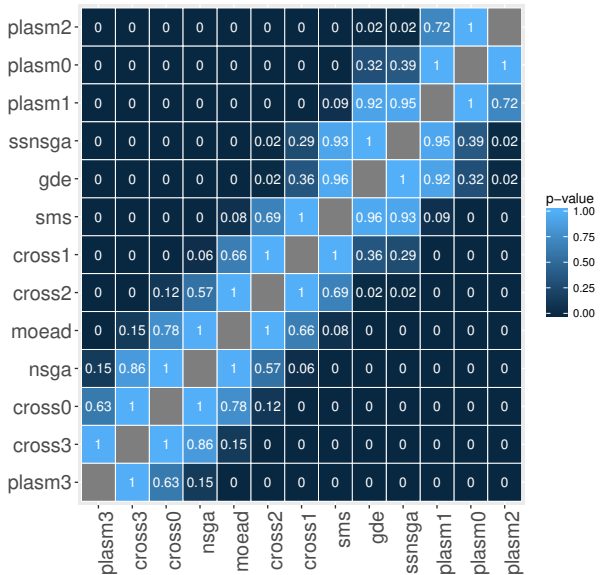


Figura 198 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_21 e $W = 20$.

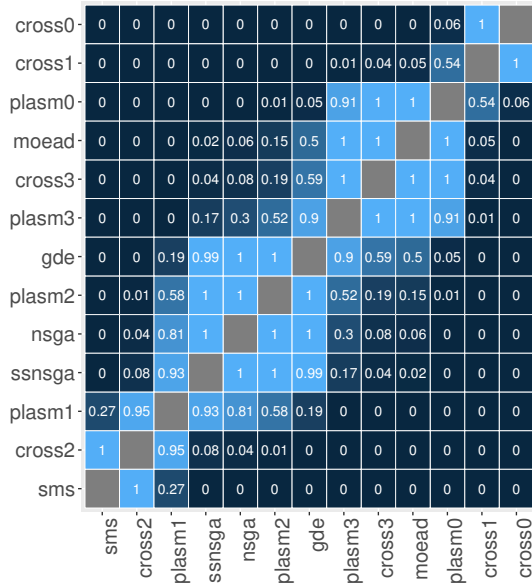


Figura 199 – *Posthoc* de Nemenyi I_{ϵ} para b60_21 e $W = 20$.

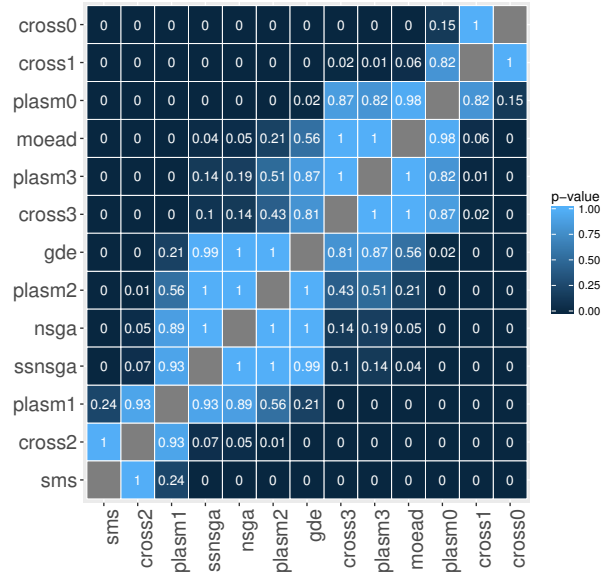


Figura 200 – *Posthoc* de Nemenyi I_H para b60_21 e $W = 20$.

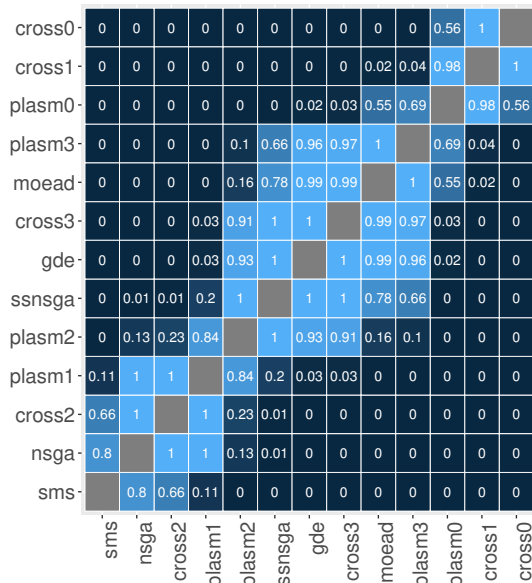
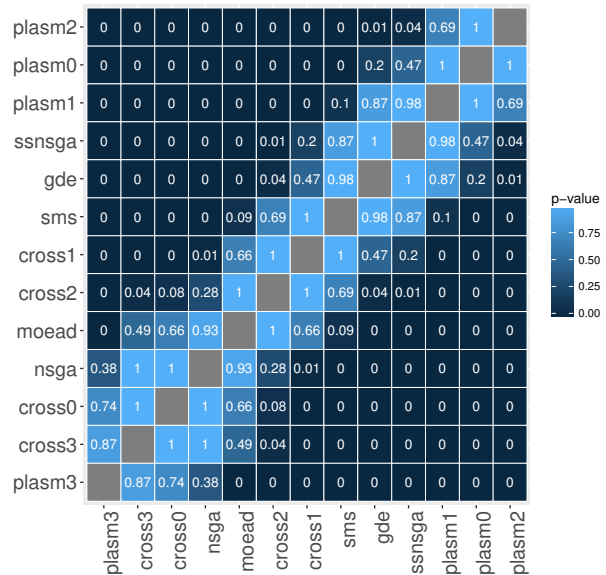


Figura 201 – *Posthoc* de Nemenyi Tempo(s) para b60_21 e $W = 20$.



D. RESULTADOS DO MODEO MULTIOBJETIVO EXPERIMENTO 2

Figura 202 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_25 e $W = 25$.

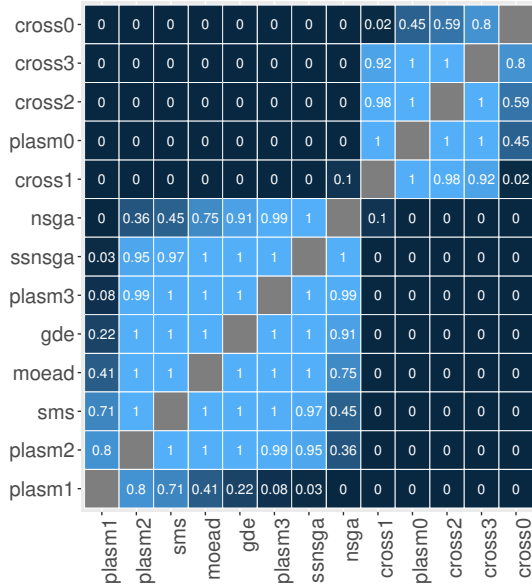


Figura 204 – *Posthoc* de Nemenyi I_H para b60_25 e $W = 25$.

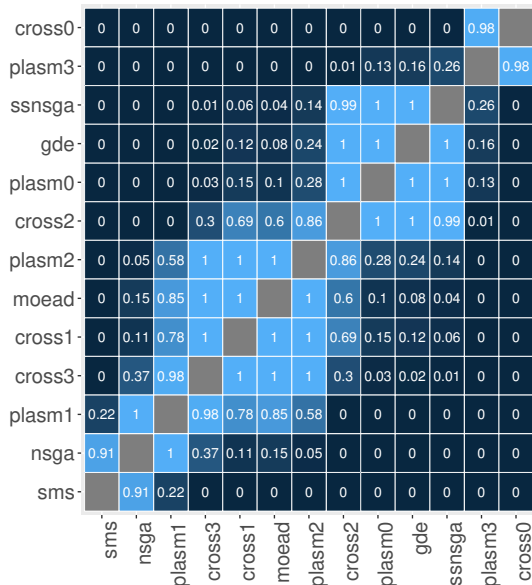


Figura 203 – *Posthoc* de Nemenyi I_{ϵ} para b60_25 e $W = 25$.

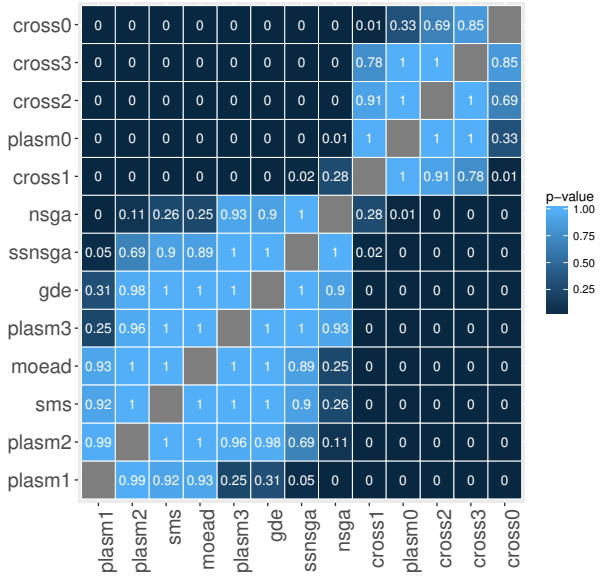


Figura 205 – *Posthoc* de Nemenyi Tempo(s) para b60_25 e $W = 25$.

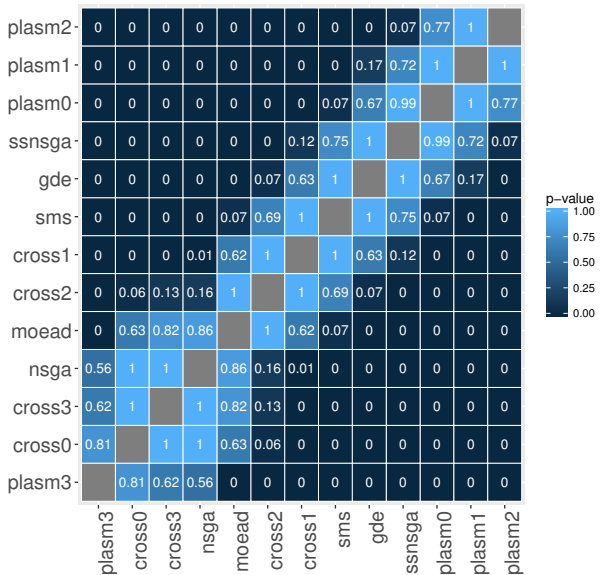


Figura 206 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_26 e $W = 25$.

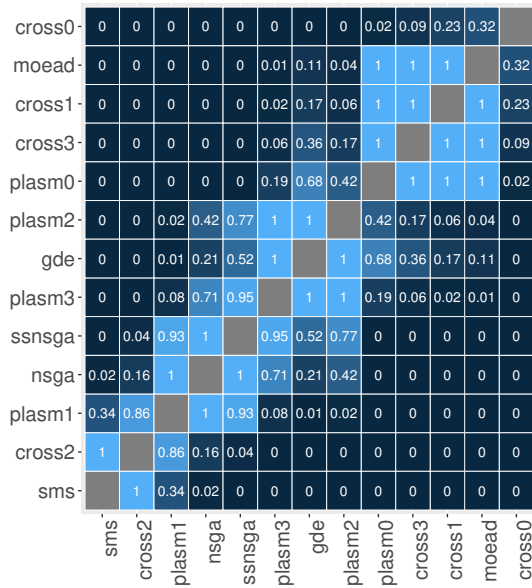


Figura 207 – *Posthoc* de Nemenyi I_{ϵ} para b60_26 e $W = 25$.

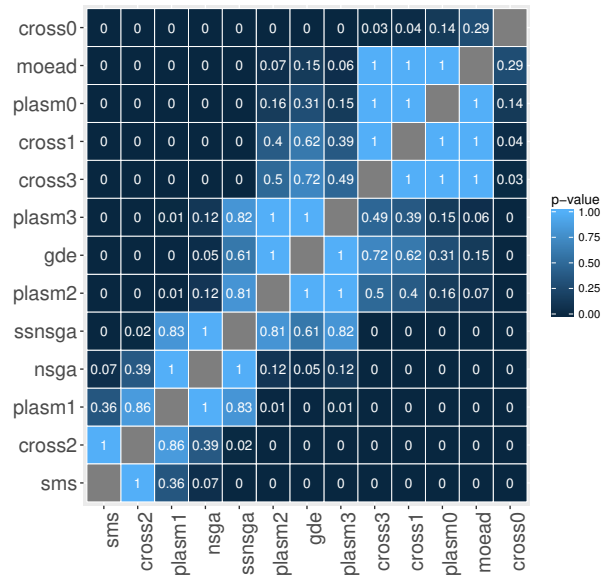


Figura 208 – *Posthoc* de Nemenyi I_H para b60_26 e $W = 25$.

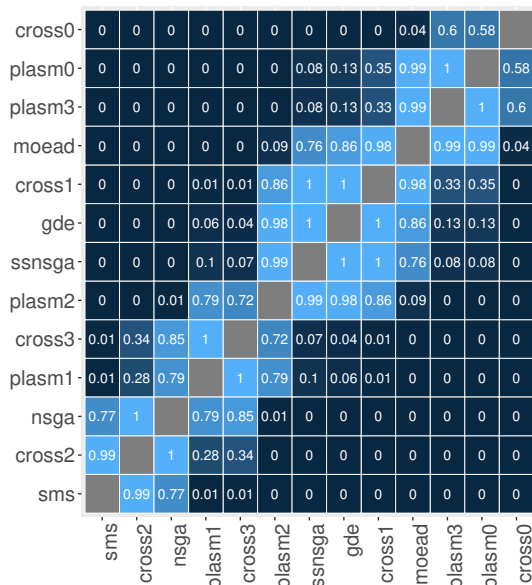
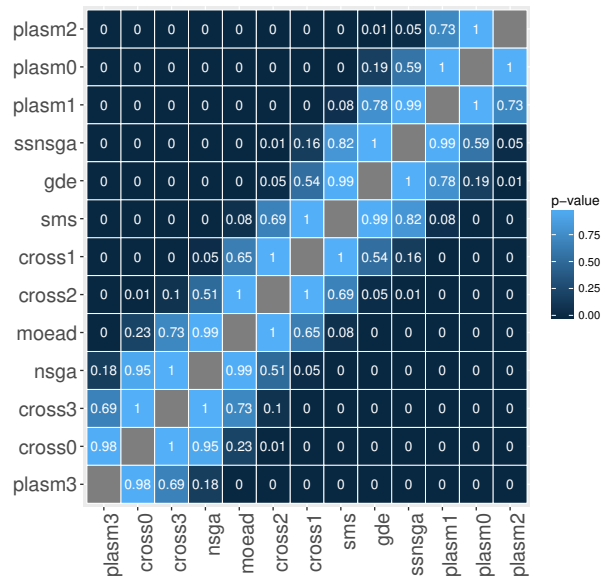


Figura 209 – *Posthoc* de Nemenyi Tempo(s) para b60_26 e $W = 25$.



D. RESULTADOS DO MODEO MULTIOBJETIVO EXPERIMENTO 2

Figura 210 – *Posthoc* de Nemenyi $I_{\epsilon+}$ para b60_27 e $W = 25$.

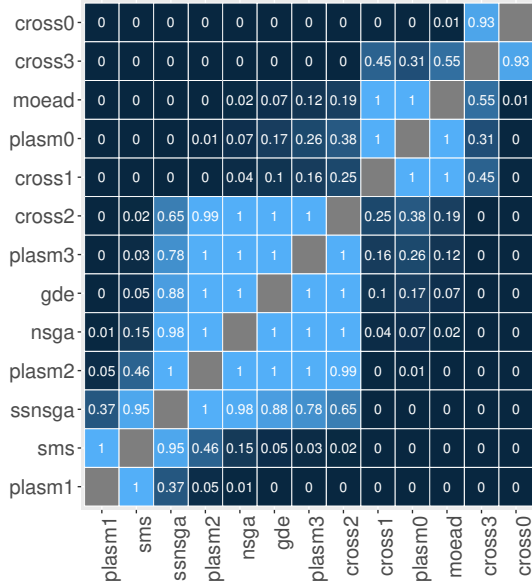


Figura 212 – *Posthoc* de Nemenyi I_H para b60_27 e $W = 25$.

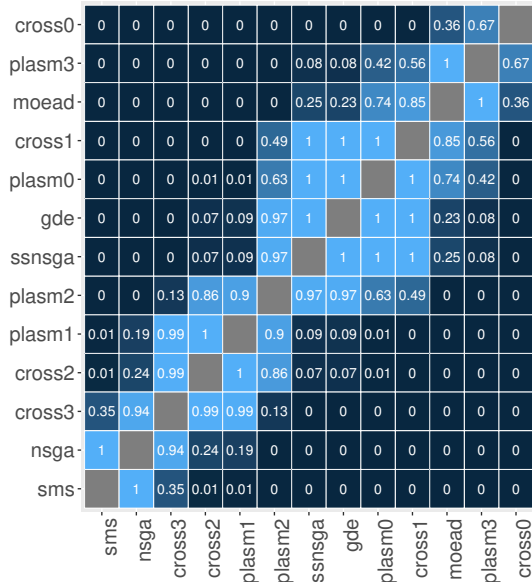


Figura 211 – *Posthoc* de Nemenyi I_{ϵ} para b60_27 e $W = 25$.

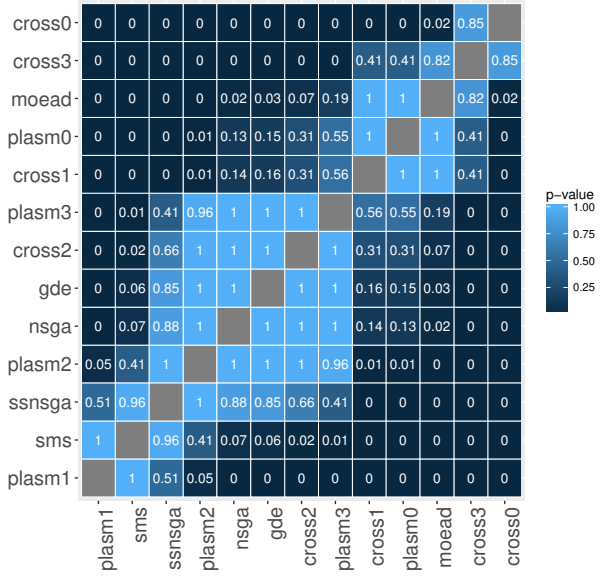


Figura 213 – *Posthoc* de Nemenyi Tempo(s) para b60_27 e $W = 25$.

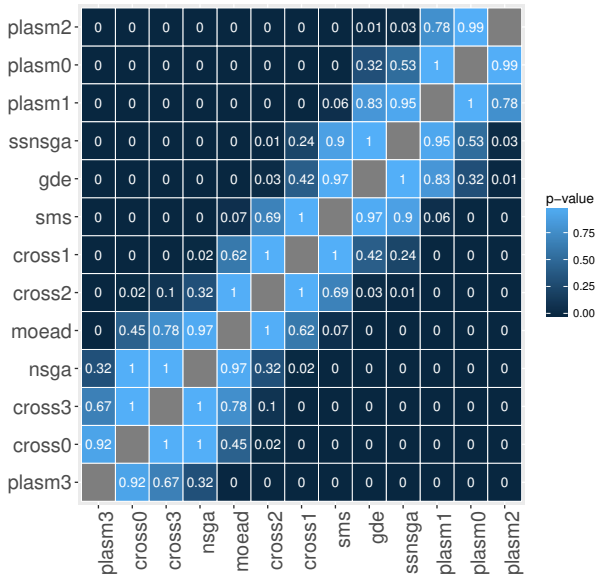


Tabela 116 – Contagem de melhores resultado no I_{ϵ_+} e $W=5$.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	2	3	5	3	6	5	2	4	5	5	6	5	51
cross1	0	0	3	4	1	4	2	0	0	5	5	3	2	29
cross2	0	2	0	1	1	3	5	3	2	2	1	3	3	26
cross3	0	2	0	0	0	2	1	1	2	1	1	2	2	14
plasm0	1	2	4	2	0	4	2	0	2	4	3	2	1	27
plasm1	0	0	0	3	0	0	0	3	3	1	3	0	0	13
plasm2	0	0	0	3	1	0	0	2	2	1	3	0	0	12
plasm3	0	2	1	3	1	3	3	0	3	3	1	3	2	25
moead	0	0	2	2	0	0	0	0	0	0	2	0	0	6
sms	0	0	0	0	0	1	2	2	4	0	2	1	1	13
nsga	0	0	0	0	0	2	2	2	3	0	0	0	1	10
ssnsga	0	0	2	2	0	0	0	1	2	2	4	0	0	13
gde	0	0	2	2	1	0	0	1	2	2	3	0	0	13
Derrotas1	10	17	27	8	25	22	17	29	26	33	20	17		

Tabela 118 – Contagem de melhores resultado no I_H e $W=5$.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	2	6	6	4	6	6	0	4	6	6	6	5	57
cross1	0	0	4	4	0	3	3	0	1	4	4	3	1	27
cross2	0	0	0	1	0	0	1	3	2	0	0	0	0	7
cross3	0	0	0	0	0	0	1	0	0	0	0	0	0	1
plasm0	0	3	3	5	0	3	2	4	3	5	3	2	2	35
plasm1	0	0	1	2	0	0	0	4	2	1	1	0	1	12
plasm2	0	0	0	2	0	0	0	1	0	0	0	0	0	3
plasm3	0	2	3	6	0	2	5	0	3	1	2	0	0	24
moead	0	0	2	3	0	0	2	1	0	0	1	0	0	9
sms	0	1	3	2	0	2	4	5	3	0	3	1	1	25
nsga	0	1	0	1	0	1	2	4	4	0	0	0	1	14
ssnsga	0	1	1	3	0	0	0	5	2	4	2	0	0	18
gde	0	1	2	4	0	1	0	5	2	5	1	0	0	21
Derrotas0	11	25	39	4	18	26	32	26	26	23	12	11		

Tabela 117 – Contagem de melhores resultado no I_{ϵ} e $W=5$.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	2	3	5	4	6	5	2	3	5	5	6	5	51
cross1	0	0	3	4	0	4	2	0	0	4	5	2	2	26
cross2	0	2	0	1	1	3	4	2	3	2	1	3	3	25
cross3	0	2	0	0	0	2	1	1	2	1	1	2	2	14
plasm0	1	2	4	2	0	2	2	1	2	4	3	2	1	26
plasm1	0	0	0	3	0	0	0	3	3	1	3	0	0	13
plasm2	0	0	0	3	1	0	0	2	1	1	3	0	0	11
plasm3	0	2	1	3	1	3	3	0	4	3	1	3	2	26
moead	0	0	2	2	0	0	0	0	0	0	2	0	0	6
sms	0	1	0	0	0	1	2	2	4	0	2	1	1	14
nsga	0	0	0	0	0	2	2	2	3	0	0	0	2	11
ssnsga	0	0	2	2	0	0	0	1	0	2	4	0	0	11
gde	0	0	2	2	1	1	0	1	2	2	2	0	0	13
Derrotas1	11	17	27	8	24	21	17	27	25	32	19	18		

Tabela 119 – Contagem de melhores resultado no Tempo e $W=5$.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	0	0	0	6	0	0	0	0	0	0	0	0	6
cross1	6	0	0	6	4	6	0	6	0	0	0	0	0	28
cross2	1	0	0	6	4	6	0	6	0	0	0	0	0	23
cross3	0	0	0	0	6	0	0	0	0	0	0	0	0	6
plasm0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasm1	6	0	0	6	2	0	0	3	0	0	0	0	0	17
plasm2	6	6	6	6	2	0	0	6	0	1	0	5	1	39
plasm3	0	0	0	0	6	3	0	0	0	0	0	0	0	9
moead	0	0	0	0	6	6	6	6	0	6	0	6	6	42
sms	6	2	6	6	0	0	0	6	0	0	0	0	0	26
nsga	0	0	0	0	5	6	6	5	0	6	0	6	6	40
ssnsga	6	0	1	6	0	3	1	6	0	0	0	0	0	23
gde	6	0	2	6	0	2	4	6	0	0	0	0	0	26
Derrotas37	8	15	42	41	32	17	50	0	13	0	17	13		

Tabela 120 – Contagem de melhores resultado no I_{ϵ_+} e $W=5$.

W=5												
	b30_01		b30_02		b30_04		b60_01		b60_02		b60_03	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	9	0	9	0	7	0	10	0	10	0	6	1
cross1	8	0	5	0	7	0	2	4	1	5	6	1
cross2	0	4	2	6	2	6	9	0	7	0	6	1
cross3	1	3	0	9	0	9	5	2	7	0	1	4
plasm0	6	0	5	0	4	2	2	3	6	2	4	1
plasm1	4	3	4	3	3	2	0	7	0	6	2	4
plasm2	1	4	2	4	1	3	5	2	1	5	2	4
plasm3	5	3	1	6	1	4	5	3	5	1	8	0
moead	0	5	2	3	2	3	1	6	1	6	0	6
sms	1	4	3	6	4	4	3	3	1	5	1	4
nsga	0	4	3	6	4	4	1	7	1	5	1	7
ssnsga	1	4	5	1	4	2	1	5	0	4	2	4
gde	1	3	4	1	2	2	2	4	2	3	2	4

Tabela 122 – Contagem de melhores resultado no I_H e $W=5$.

W=5												
	b30_01		b30_02		b30_04		b60_01		b60_02		b60_03	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	9	0	9	0	9	0	10	0	11	0	9	0
cross1	8	0	7	0	4	1	0	4	0	6	8	0
cross2	0	4	1	5	2	6	0	4	1	3	3	3
cross3	0	5	0	9	0	9	0	4	0	3	1	9
plasm0	9	0	5	1	5	1	8	0	4	1	4	1
plasm1	1	4	5	2	3	3	0	3	0	3	3	3
plasm2	0	4	1	6	2	4	0	4	0	2	0	6
plasm3	4	5	2	7	1	8	8	2	7	3	2	7
moead	0	4	3	3	3	3	0	5	1	6	2	5
sms	1	4	3	6	5	3	5	5	6	3	5	5
nsga	1	3	3	6	5	4	1	3	1	4	3	3
ssnsga	1	3	4	2	5	1	2	2	2	2	4	2
gde	4	2	5	1	2	3	3	1	4	1	3	3

Tabela 121 – Contagem de melhores resultado no I_{ϵ} e $W=5$.

W=5												
	b30_01		b30_02		b30_04		b60_01		b60_02		b60_03	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	9	0	9	0	7	0	10	0	10	0	6	1
cross1	8	0	5	0	5	0	1	5	1	5	6	1
cross2	1	4	2	6	1	6	8	0	7	0	6	1
cross3	1	3	0	9	0	9	5	2	7	0	1	4
plasm0	6	0	5	1	4	1	1	3	6	2	4	1
plasm1	4	4	5	2	2	2	0	6	0	6	2	4
plasm2	0	4	2	4	1	2	5	2	1	5	2	4
plasm3	6	3	1	7	1	4	5	2	5	1	8	0
moead	0	5	2	3	2	2	1	6	1	6	0	5
sms	1	4	3	6	4	4	4	2	1	5	1	4
nsga	0	4	4	5	4	4	1	7	1	5	1	7
ssnsga	0	4	5	1	4	1	1	5	0	4	1	4
gde	2	3	3	2	2	2	2	4	2	3	2	4

Tabela 123 – Contagem de melhores resultado no Tempo e $W=5$.

W=5												
	b30_01		b30_02		b30_04		b60_01		b60_02		b60_03	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	1	6	1	6	1	6	1	6	1	7	1	6
cross1	5	2	5	1	4	2	5	1	4	1	5	1
cross2	4	2	4	2	3	3	4	2	4	4	4	2
cross3	1	7	1	7	1	7	1	7	1	7	1	7
plasm0	0	7	0	7	0	7	0	7	0	6	0	7
plasm1	3	5	3	6	3	5	3	4	3	5	2	7
plasm2	6	3	6	3	8	2	7	3	6	3	6	3
plasm3	1	8	1	9	2	8	1	9	2	8	2	8
moead	7	0	7	0	7	0	7	0	7	0	7	0
sms	5	2	4	2	5	2	4	3	4	2	4	2
nsga	6	0	7	0	7	0	7	0	6	0	7	0
ssnsga	4	3	4	3	3	3	3	3	5	2	4	3
gde	4	2	5	2	4	3	4	2	4	2	5	2

Tabela 124 – Contagem de melhores resultado no I_{ϵ_+} e W=10.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	0	5	5	2	6	5	3	1	5	5	6	5	48
cross1	0	0	5	5	3	6	6	3	1	5	5	6	5	50
cross2	0	0	0	0	0	1	4	0	0	0	0	1	1	7
cross3	0	0	1	0	0	2	4	2	1	1	0	2	0	13
plasm0	0	0	5	3	0	1	3	0	2	5	3	4	1	27
plasm1	0	0	2	0	3	0	1	4	6	2	2	0	1	21
plasm2	0	0	1	0	0	0	0	0	1	1	1	0	0	4
plasm3	0	0	4	1	0	1	1	0	0	1	0	1	1	10
moead	0	0	4	2	0	0	1	0	0	0	0	0	0	7
sms	0	0	0	2	0	0	4	4	5	0	0	0	1	16
nsga	0	0	0	0	1	0	2	4	4	1	0	0	2	14
ssnsga	0	0	3	1	0	0	0	2	4	5	2	0	0	17
gde	0	1	4	4	2	1	0	0	1	3	1	0	0	17
Derrotas0	1	34	23	11	18	31	22	26	29	19	20	17		

Tabela 126 – Contagem de melhores resultado no I_H e W=10.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	1	5	5	4	6	6	2	3	6	6	6	6	56
cross1	0	0	5	5	1	5	5	0	1	6	5	5	5	43
cross2	0	1	0	0	0	1	4	0	2	1	0	1	1	11
cross3	0	1	1	0	0	1	1	2	3	2	2	1	1	15
plasm0	0	0	6	6	0	5	5	1	0	5	4	2	0	34
plasm1	0	0	1	2	1	0	0	5	5	3	2	0	0	19
plasm2	0	0	0	0	0	0	0	1	1	1	1	0	1	5
plasm3	0	1	4	1	0	1	4	0	1	1	2	2	1	18
moead	0	0	3	1	0	0	5	0	0	0	0	0	0	9
sms	0	0	0	1	0	0	3	4	5	0	1	0	0	14
nsga	0	0	0	0	0	0	1	4	5	0	0	0	0	10
ssnsga	0	0	3	0	0	0	0	3	3	5	2	0	0	16
gde	0	0	3	1	2	1	0	3	1	5	2	0	0	18
Derrotas0	4	31	22	8	20	34	25	30	35	27	17	15		

Tabela 125 – Contagem de melhores resultado no I_{ϵ} e W=10.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	0	5	5	2	6	5	2	1	5	5	6	5	47
cross1	0	0	5	5	3	6	6	3	2	5	5	6	5	51
cross2	0	0	0	0	0	1	4	0	0	0	0	1	1	7
cross3	0	0	1	0	0	2	4	2	1	1	0	2	0	13
plasm0	0	0	5	4	0	1	3	0	0	5	4	3	0	25
plasm1	0	0	2	0	3	0	0	4	6	2	2	0	1	20
plasm2	0	0	1	0	0	0	0	0	1	1	1	0	0	4
plasm3	0	0	4	1	0	1	1	0	0	1	0	1	1	10
moead	0	0	4	2	0	0	1	0	0	0	0	0	0	7
sms	0	0	0	2	0	0	4	4	5	0	0	0	1	16
nsga	0	0	0	0	1	0	2	4	4	1	0	0	2	14
ssnsga	0	0	4	1	0	0	0	2	4	5	2	0	0	18
gde	0	1	4	2	3	1	0	0	0	3	2	0	0	16
Derrotas0	1	35	22	12	18	30	21	24	29	21	19	16		

Tabela 127 – Contagem de melhores resultado no Tempo e W=10.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	0	0	0	6	0	0	0	0	0	0	0	0	6
cross1	6	0	0	6	6	6	0	6	0	0	0	0	0	30
cross2	1	0	0	5	6	6	0	6	0	0	0	0	0	24
cross3	0	0	0	0	6	0	0	0	0	0	0	0	0	6
plasm0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasm1	6	0	0	6	0	0	0	3	0	0	0	0	0	15
plasm2	6	6	6	6	0	0	0	6	0	0	0	0	0	30
plasm3	0	0	0	0	6	3	0	0	0	0	0	0	0	9
moead	0	0	0	0	6	6	6	6	0	5	0	6	4	39
sms	6	0	0	6	3	4	6	6	0	0	0	0	0	31
nsga	0	1	0	0	6	6	6	3	0	6	0	6	6	40
ssnsga	6	0	6	6	0	0	6	6	0	0	0	0	0	30
gde	6	0	6	6	0	0	5	6	2	0	0	0	0	31
Derrotas37	7	18	41	45	31	29	48	2	11	0	12	10		

Tabela 128 – Contagem de melhores resultado no I_{ϵ_+} e $W=10$.

W=10												
	b30_07		b30_08		b30_09		b60_07		b60_08		b60_09	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	4	0	9	0	9	0	10	0	8	0	8	0
cross1	3	1	9	0	9	0	10	0	10	0	9	0
cross2	4	0	1	6	0	9	0	4	1	8	1	7
cross3	3	1	1	6	1	4	4	3	2	5	2	4
plasm0	4	0	4	2	6	2	5	2	3	3	5	2
plasm1	3	6	2	2	3	2	5	4	3	2	5	2
plasm2	3	6	0	6	1	4	0	4	0	5	0	6
plasm3	4	0	2	4	1	5	1	3	1	5	1	5
moead	0	3	2	5	1	3	0	5	2	4	2	6
sms	0	3	3	5	4	5	1	6	4	5	4	5
nsga	0	4	4	2	2	3	1	4	5	3	2	3
ssnsga	3	6	3	2	2	3	1	4	4	2	4	3
gde	3	4	3	3	3	2	3	2	2	3	3	3

Tabela 130 – Contagem de melhores resultado no I_H e $W=10$.

W=10												
	b30_07		b30_08		b30_09		b60_07		b60_08		b60_09	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	9	0	9	0	9	0	10	0	10	0	9	0
cross1	1	4	9	0	8	0	9	0	8	0	8	0
cross2	6	1	1	4	1	8	1	4	1	7	1	7
cross3	7	1	1	4	3	4	1	4	1	5	2	4
plasm0	5	1	5	2	7	1	7	1	4	2	6	1
plasm1	2	6	3	3	3	3	4	3	4	2	3	3
plasm2	5	4	0	5	0	7	0	5	0	6	0	7
plasm3	6	2	6	2	2	4	1	6	1	5	2	6
moead	0	5	1	7	2	4	1	5	3	4	2	5
sms	0	5	2	5	3	6	2	7	3	6	4	6
nsga	0	6	1	5	2	4	2	3	3	5	2	4
ssnsga	2	4	1	3	3	3	2	3	4	2	4	2
gde	1	5	3	2	3	2	3	2	4	2	4	2

Tabela 129 – Contagem de melhores resultado no I_{ϵ} e $W=10$.

W=10												
	b30_07		b30_08		b30_09		b60_07		b60_08		b60_09	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	4	0	9	0	8	0	10	0	8	0	8	0
cross1	3	1	9	0	10	0	10	0	10	0	9	0
cross2	4	0	1	7	0	9	0	4	1	8	1	7
cross3	3	0	1	6	1	4	4	2	2	6	2	4
plasm0	3	1	4	2	6	2	5	2	3	3	4	2
plasm1	3	6	2	2	3	2	5	4	3	2	4	2
plasm2	3	6	0	6	1	4	0	4	0	5	0	5
plasm3	4	0	2	4	1	4	1	3	1	5	1	5
moead	0	3	2	4	1	4	0	5	2	3	2	5
sms	0	3	3	5	4	5	1	6	4	5	4	5
nsga	0	4	4	3	2	3	1	4	5	3	2	4
ssnsga	3	6	4	2	2	3	1	4	4	2	4	2
gde	3	3	3	3	3	2	2	2	2	3	3	3

Tabela 131 – Contagem de melhores resultado no Tempo e $W=10$.

W=10												
	b30_07		b30_08		b30_09		b60_07		b60_08		b60_09	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	1	6	1	6	1	6	1	7	1	6	1	6
cross1	5	1	5	1	5	1	5	2	5	1	5	1
cross2	4	3	4	3	4	3	4	3	4	3	4	3
cross3	1	7	1	7	1	7	1	6	1	7	1	7
plasm0	0	7	0	7	0	7	0	8	0	8	0	8
plasm1	3	5	2	6	2	6	3	4	3	4	2	6
plasm2	5	4	5	5	5	5	5	5	5	5	5	5
plasm3	1	8	2	8	2	8	1	8	1	8	2	8
moead	7	0	7	0	7	0	6	1	5	1	7	0
sms	5	2	5	2	5	2	5	2	5	1	6	2
nsga	6	0	7	0	7	0	7	0	6	0	7	0
ssnsga	5	2	5	2	5	2	5	2	5	2	5	2
gde	4	2	5	2	5	2	6	1	6	1	5	2

Tabela 132 – Contagem de melhores resultado no I_{ϵ_+} e W=15.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	2	4	3	2	6	5	5	3	6	6	5	5	52
cross1	0	0	4	3	1	6	6	3	0	6	4	6	5	44
cross2	0	0	0	0	1	2	4	1	2	2	0	2	1	15
cross3	0	0	2	0	0	2	5	1	1	4	3	3	3	24
plasm0	0	0	4	2	0	1	5	0	1	4	4	5	1	27
plasm1	0	0	0	2	4	0	1	4	6	1	2	0	0	20
plasm2	0	0	0	0	0	0	0	0	2	0	2	0	0	4
plasm3	0	0	4	1	0	1	2	0	0	2	0	2	1	13
moead	0	0	3	2	0	0	2	0	0	0	0	0	0	7
sms	0	0	1	0	1	0	4	3	5	0	1	0	1	16
nsga	0	0	0	0	1	0	1	3	4	2	0	1	1	13
ssnsga	0	0	2	2	0	0	0	1	4	3	2	0	0	14
gde	0	1	3	2	4	2	0	1	3	3	2	0	0	21
Derrotas0	3	27	17	14	20	35	22	31	33	26	24	18		

Tabela 134 – Contagem de melhores resultado no I_H e W=15.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	1	5	5	2	6	6	4	5	6	6	6	6	58
cross1	0	0	4	3	0	6	6	0	0	6	5	6	4	40
cross2	0	0	0	0	0	2	4	3	0	2	1	2	1	15
cross3	0	0	1	0	0	1	4	2	4	3	2	1	0	18
plasm0	0	0	4	3	0	4	6	0	1	6	5	5	1	35
plasm1	0	0	0	2	2	0	1	5	6	0	1	2	1	20
plasm2	0	0	0	0	0	0	0	0	1	0	0	0	0	1
plasm3	0	0	1	2	0	1	4	0	0	2	1	2	0	13
moead	0	0	3	0	0	0	4	0	0	0	0	0	0	7
sms	0	0	1	1	0	0	4	4	5	0	1	1	0	17
nsga	0	0	0	0	0	0	3	4	4	1	0	0	0	12
ssnsga	0	0	2	0	0	0	0	2	2	3	3	0	0	12
gde	0	2	3	2	4	2	0	3	0	4	2	0	0	22
Derrotas0	3	24	18	8	22	42	27	28	33	27	25	13		

Tabela 133 – Contagem de melhores resultado no I_{ϵ} e W=15.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	2	5	4	2	6	5	4	3	6	5	5	5	52
cross1	0	0	4	3	1	6	5	3	0	6	4	6	4	42
cross2	0	0	0	0	1	2	4	1	2	2	0	2	1	15
cross3	0	0	2	0	0	2	4	1	1	4	3	3	2	22
plasm0	0	0	4	1	0	2	4	0	1	4	4	5	1	26
plasm1	0	0	1	2	3	0	1	4	6	1	3	0	0	21
plasm2	0	0	0	0	0	0	0	0	2	0	1	0	0	3
plasm3	0	0	4	1	0	1	2	0	0	2	0	2	1	13
moead	0	0	3	2	0	0	2	0	0	0	0	0	0	7
sms	0	0	1	0	1	0	4	4	5	0	1	1	1	18
nsga	0	0	0	0	1	0	2	3	4	1	0	1	1	13
ssnsga	0	0	2	2	0	0	0	1	5	3	1	0	0	14
gde	0	2	3	3	4	0	0	1	3	3	2	0	0	21
Derrotas0	4	29	18	13	19	33	22	32	32	24	25	16		

Tabela 135 – Contagem de melhores resultado no Tempo e W=15.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	0	0	0	6	0	0	0	0	0	0	0	0	6
cross1	6	0	0	6	6	6	0	6	0	0	0	0	0	30
cross2	1	0	0	6	6	6	0	6	0	0	0	0	0	25
cross3	0	0	0	0	6	0	0	0	0	0	0	0	0	6
plasm0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasm1	6	0	0	6	0	0	0	5	0	0	0	0	0	17
plasm2	6	6	6	6	0	0	0	6	0	0	0	0	0	30
plasm3	0	0	0	0	6	1	0	0	0	0	0	0	0	7
moead	0	0	0	0	6	6	6	5	0	5	0	6	5	39
sms	6	0	0	6	2	5	6	6	0	0	0	0	0	31
nsga	0	0	0	0	6	6	6	3	0	5	0	6	6	38
ssnsga	6	0	6	6	0	0	6	6	0	0	0	0	0	30
gde	6	0	6	6	0	0	6	6	1	0	0	0	0	31
Derrotas37	6	18	42	44	30	30	49	1	10	0	12	11		

Tabela 136 – Contagem de melhores resultado no I_{ϵ_+} e $W=15$.

W=15												
	b30_13		b30_14		b30_15		b60_13		b60_14		b60_15	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	10	0	9	0	6	0	9	0	10	0	8	0
cross1	8	0	4	2	10	0	9	0	5	1	8	0
cross2	1	5	4	1	1	7	1	8	8	0	0	6
cross3	1	7	7	0	2	6	3	3	5	0	6	1
plasm0	6	2	6	1	2	4	5	2	3	3	5	2
plasm1	3	3	3	5	3	2	4	3	3	5	4	2
plasm2	0	5	2	6	0	6	0	6	2	6	0	6
plasm3	2	5	3	3	1	4	1	5	4	2	2	3
moead	1	6	0	7	2	3	2	3	0	7	2	5
sms	4	6	0	6	5	4	3	6	2	4	2	7
nsga	3	3	0	7	6	2	2	4	0	6	2	4
ssnsga	4	3	2	6	3	2	2	3	2	6	1	4
gde	4	2	6	2	2	3	4	2	2	6	3	3

Tabela 138 – Contagem de melhores resultado no I_H e $W=15$.

W=15												
	b30_13		b30_14		b30_15		b60_13		b60_14		b60_15	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	9	0	9	0	9	0	11	0	10	0	10	0
cross1	7	1	5	2	8	0	8	0	5	0	7	0
cross2	1	4	5	1	2	6	1	7	5	1	1	5
cross3	2	5	6	1	2	3	2	4	2	3	4	2
plasm0	7	1	6	1	6	2	5	2	5	1	6	1
plasm1	4	4	2	5	3	3	5	2	3	5	3	3
plasm2	0	7	1	6	0	7	0	7	0	8	0	7
plasm3	2	5	4	3	0	6	1	5	5	1	1	7
moead	0	5	0	5	2	5	2	5	1	4	2	4
sms	4	6	0	6	3	5	5	4	2	6	3	6
nsga	3	3	0	6	3	5	3	5	1	4	2	4
ssnsga	2	4	0	6	4	3	2	4	2	5	2	3
gde	5	1	5	1	5	2	3	3	1	4	3	2

Tabela 137 – Contagem de melhores resultado no I_{ϵ} e $W=15$.

W=15												
	b30_13		b30_14		b30_15		b60_13		b60_14		b60_15	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	9	0	10	0	6	0	9	0	10	0	8	0
cross1	7	1	4	2	9	0	9	0	5	1	8	0
cross2	1	5	4	2	1	8	1	8	8	0	0	6
cross3	1	7	7	0	2	5	3	3	5	1	4	2
plasm0	7	1	6	1	1	4	5	2	3	3	4	2
plasm1	2	3	3	5	5	2	4	2	3	5	4	2
plasm2	0	5	1	7	0	5	0	6	2	6	0	4
plasm3	2	4	3	3	1	4	1	5	5	2	1	4
moead	1	6	0	7	2	3	2	4	0	7	2	5
sms	4	6	0	6	6	3	3	6	2	5	3	6
nsga	3	3	1	5	6	3	2	4	0	5	1	4
ssnsga	4	3	1	6	2	3	3	3	2	6	2	4
gde	4	1	6	2	2	3	3	2	2	6	4	2

Tabela 139 – Contagem de melhores resultado no Tempo e $W=15$.

W=15												
	b30_13		b30_14		b30_15		b60_13		b60_14		b60_15	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	1	6	1	6	1	7	1	6	1	6	1	6
cross1	5	1	5	1	5	1	5	1	5	1	5	1
cross2	4	3	4	3	5	3	4	3	4	3	4	3
cross3	1	7	1	7	1	7	1	7	1	7	1	7
plasm0	0	7	0	7	0	7	0	8	0	7	0	8
plasm1	3	5	2	6	3	5	3	5	3	5	3	4
plasm2	5	5	5	5	5	5	5	5	5	5	5	5
plasm3	1	9	2	7	1	9	1	8	1	8	1	8
moead	7	0	6	0	7	0	7	0	6	0	6	1
sms	5	2	5	1	5	2	6	2	5	1	5	2
nsga	7	0	6	0	7	0	6	0	6	0	6	0
ssnsga	5	2	5	2	5	2	5	2	5	2	5	2
gde	5	2	5	2	5	2	5	2	5	2	6	1

Tabela 140 – Contagem de melhores resultado no I_{ϵ_+} e W=20.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	0	5	4	2	6	6	6	3	6	6	6	6	56
cross1	0	0	4	4	1	6	6	4	1	6	4	6	4	46
cross2	0	0	0	1	1	2	4	1	0	1	1	2	1	14
cross3	0	0	4	0	0	2	2	2	0	4	3	4	2	23
plasm0	0	0	4	2	0	1	6	1	0	5	3	5	1	28
plasm1	0	0	0	2	5	0	0	6	6	2	2	1	0	24
plasm2	0	0	0	0	0	0	0	1	2	1	1	0	0	5
plasm3	0	0	4	0	1	0	0	0	0	2	1	1	0	9
moead	0	0	4	0	0	0	3	0	0	0	0	0	0	7
sms	0	0	0	1	0	0	4	2	4	0	1	0	0	12
nsga	0	0	2	0	0	0	0	2	3	2	0	0	0	9
ssnsga	0	0	1	0	1	0	0	0	5	5	1	0	0	13
gde	0	2	4	0	5	1	0	1	3	5	2	0	0	23
Derrotas0	2	32	14	16	18	31	26	27	39	25	25	14		

Tabela 142 – Contagem de melhores resultado no I_H e W=20.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	0	6	4	2	6	6	5	5	6	6	6	6	58
cross1	0	0	4	4	0	6	6	3	1	6	6	6	3	45
cross2	0	0	0	1	0	1	4	2	0	1	1	2	1	13
cross3	0	0	3	0	0	2	2	3	2	3	3	2	2	22
plasm0	0	0	4	3	0	3	5	1	0	4	6	4	1	31
plasm1	0	0	1	2	3	0	0	5	5	2	2	1	0	21
plasm2	0	0	0	0	0	0	0	1	2	1	0	0	0	4
plasm3	0	0	2	0	0	0	2	0	0	1	1	1	0	7
moead	0	0	4	1	0	0	1	0	0	0	0	0	0	6
sms	0	0	0	1	1	0	4	4	5	0	1	1	0	17
nsga	0	0	0	0	0	0	1	4	5	0	0	0	0	10
ssnsga	0	0	4	1	1	0	0	1	3	3	3	0	0	16
gde	0	2	5	1	4	3	0	1	2	4	3	0	0	25
Derrotas0	2	33	18	11	21	31	30	30	31	32	23	13		

Tabela 141 – Contagem de melhores resultado no I_{ϵ} e W=20.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	1	5	4	2	6	6	6	4	6	6	6	6	58
cross1	0	0	4	4	1	5	6	4	1	6	4	6	4	45
cross2	0	0	0	1	0	2	4	1	0	1	1	2	1	13
cross3	0	0	4	0	0	2	2	2	1	4	3	3	2	23
plasm0	0	0	4	2	0	1	6	1	0	5	4	5	1	29
plasm1	0	1	0	2	5	0	0	6	6	1	2	1	0	24
plasm2	0	0	0	0	0	0	0	1	2	1	1	0	0	5
plasm3	0	0	4	0	1	0	0	0	0	1	0	0	0	6
moead	0	0	4	0	0	0	1	0	0	0	0	0	0	5
sms	0	0	0	1	0	0	4	3	4	0	0	0	0	12
nsga	0	0	2	0	0	0	0	3	3	3	0	0	0	11
ssnsga	0	0	1	0	1	0	0	1	5	5	2	0	0	15
gde	0	1	4	0	5	1	0	1	3	5	2	0	0	22
Derrotas0	3	32	14	15	17	29	29	29	38	25	23	14		

Tabela 143 – Contagem de melhores resultado no Tempo e W=20.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moead	sms	nsga	ssnsga	gde	Total
cross0	0	0	0	0	6	0	0	0	0	0	0	0	0	6
cross1	6	0	0	6	6	6	0	6	0	0	0	0	0	30
cross2	1	0	0	6	6	6	0	6	0	0	0	0	0	25
cross3	0	0	0	0	6	0	0	0	0	0	0	0	0	6
plasm0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasm1	6	0	0	6	0	0	0	3	0	0	0	0	0	15
plasm2	6	6	6	6	0	0	0	6	0	0	0	0	0	30
plasm3	0	0	0	0	6	3	0	0	0	0	0	0	0	9
moead	0	0	0	0	6	6	6	4	0	3	0	6	3	34
sms	6	0	0	6	3	3	6	6	0	0	0	0	0	30
nsga	0	1	0	0	6	6	6	3	0	4	0	6	6	38
ssnsga	6	0	6	6	0	0	6	6	0	0	0	0	0	30
gde	6	0	6	6	0	0	6	6	3	0	0	0	0	33
Derrotas37	7	18	42	45	30	30	46	3	7	0	12	9		

Tabela 144 – Contagem de melhores resultado no I_{ϵ_+} e $W=20$.

W=20												
	b30_19		b30_20		b30_21		b60_19		b60_20		b60_21	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	9	0	9	0	10	0	9	0	9	0	10	0
cross1	8	1	5	1	8	0	10	0	6	0	9	0
cross2	1	8	5	2	7	0	0	7	0	7	1	8
cross3	1	4	7	0	1	4	3	3	8	0	3	3
plasm0	6	3	3	3	4	3	6	2	4	3	5	2
plasm1	4	3	5	4	4	4	3	2	4	3	4	2
plasm2	0	6	2	5	3	4	0	5	0	6	0	5
plasm3	1	6	1	5	2	4	1	5	2	3	2	3
moead	2	3	0	5	0	6	2	5	2	4	1	4
sms	4	5	0	6	0	7	3	6	3	7	2	8
nsga	2	4	0	5	2	4	2	5	1	4	2	3
ssnsga	3	3	2	5	2	6	2	3	2	4	2	4
gde	6	1	4	2	3	4	4	2	3	3	3	2

Tabela 146 – Contagem de melhores resultado no I_H e $W=20$.

W=20												
	b30_19		b30_20		b30_21		b60_19		b60_20		b60_21	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	10	0	9	0	9	0	11	0	9	0	10	0
cross1	8	1	7	0	7	0	7	0	6	1	10	0
cross2	1	8	4	2	5	2	2	6	0	8	1	7
cross3	2	4	7	0	2	4	1	6	7	0	3	4
plasm0	5	4	4	2	5	1	5	1	5	2	7	1
plasm1	4	3	4	4	5	3	2	4	3	3	3	4
plasm2	0	5	1	6	3	4	0	5	0	7	0	4
plasm3	1	6	3	4	1	5	0	5	2	4	0	6
moead	1	5	0	6	0	6	2	3	2	5	1	5
sms	6	3	2	6	0	4	3	6	3	6	3	6
nsga	2	5	0	6	1	5	3	6	2	4	2	6
ssnsga	3	4	1	6	2	4	4	2	3	4	3	3
gde	6	1	3	3	2	4	5	1	4	2	5	2

Tabela 145 – Contagem de melhores resultado no I_{ϵ} e $W=20$.

W=20													
		b30_19		b30_20		b30_21		b60_19		b60_20		b60_21	
		v	d	v	d	v	d	v	d	v	d	v	d
cross0	10	0	10	0	10	0	9	0	9	0	10	0	
cross1	8	0	5	2	8	0	10	0	5	1	9	0	
cross2	1	8	5	2	6	0	0	7	0	7	1	8	
cross3	2	4	7	0	1	4	3	3	8	0	2	3	
plasm0	6	3	4	3	4	2	6	2	4	3	5	2	
plasm1	4	3	4	4	4	4	3	2	5	2	4	2	
plasm2	0	5	2	5	3	4	0	4	0	6	0	5	
plasm3	1	6	1	5	0	6	1	5	2	3	1	4	
moead	1	5	0	5	0	6	1	5	2	4	1	4	
sms	4	5	0	5	0	7	3	6	2	8	3	7	
nsga	2	5	0	6	3	3	2	5	2	3	2	3	
ssnsga	4	3	2	5	3	5	2	3	2	4	2	3	
gde	5	1	4	2	3	4	4	2	3	3	3	2	

Tabela 147 – Contagem de melhores resultado no Tempo e $W=20$.

W=20												
	b30_19		b30_20		b30_21		b60_19		b60_20		b60_21	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	1	6	1	6	1	6	1	7	1	6	1	6
cross1	5	1	5	1	5	1	5	1	5	1	5	2
cross2	4	3	4	3	4	3	5	3	4	3	4	3
cross3	1	7	1	7	1	7	1	7	1	7	1	7
plasm0	0	7	0	7	0	7	0	8	0	8	0	8
plasm1	2	6	3	5	2	6	2	5	3	4	3	4
plasm2	5	5	5	5	5	5	5	5	5	5	5	5
plasm3	2	7	1	8	2	8	2	7	1	8	1	8
moead	6	0	6	0	7	0	5	1	5	1	5	1
sms	5	1	5	1	5	2	5	1	5	1	5	1
nsga	6	0	6	0	7	0	6	0	6	0	7	0
ssnsga	5	2	5	2	5	2	5	2	5	2	5	2
gde	5	2	5	2	5	2	6	1	6	1	6	1

Tabela 148 – Contagem de melhores resultado no I_{ϵ_+} e W=25.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moad	sms	nsga	ssnsga	gde	Total
cross0	0	2	5	2	3	6	6	6	5	6	6	6	6	59
cross1	0	0	3	2	0	4	5	3	2	6	4	6	2	37
cross2	0	0	0	0	0	2	5	2	1	3	1	2	1	17
cross3	0	0	2	0	0	3	4	3	1	4	3	4	2	26
plasm0	0	0	3	2	0	2	5	1	1	5	3	4	0	26
plasm1	0	2	1	1	4	0	2	1	5	1	3	0	2	22
plasm2	0	0	0	0	0	0	0	0	1	0	0	0	0	1
plasm3	0	0	2	0	1	1	1	0	1	2	0	1	0	9
moad	0	0	3	2	0	0	1	0	0	0	0	0	0	6
sms	0	0	0	0	1	0	3	3	5	0	0	0	0	12
nsga	0	0	0	0	1	0	3	2	4	2	0	0	0	12
ssnsga	0	0	2	1	2	1	0	0	2	3	3	0	0	14
gde	0	1	4	2	4	1	0	1	0	4	2	0	0	19
Derrotas0	5	25	12	16	20	35	22	28	36	25	23	13		

Tabela 150 – Contagem de melhores resultado no I_H e W=25.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moad	sms	nsga	ssnsga	gde	Total
cross0	0	4	6	6	2	6	6	2	5	6	6	6	6	61
cross1	0	0	5	5	0	3	2	1	0	5	5	2	2	30
cross2	0	0	0	0	0	1	3	4	0	1	1	1	0	11
cross3	0	0	0	0	0	0	2	5	0	1	0	1	0	9
plasm0	0	0	5	6	0	5	4	0	1	6	6	2	0	35
plasm1	0	2	1	1	1	0	0	6	4	3	1	0	0	19
plasm2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasm3	0	0	1	0	0	0	4	0	0	1	1	1	0	8
moad	0	0	4	4	0	0	2	1	0	0	0	0	0	11
sms	0	1	2	2	0	1	5	5	6	0	1	2	1	26
nsga	0	0	0	0	0	0	6	5	4	0	0	0	0	15
ssnsga	0	0	3	4	0	2	0	0	2	3	5	0	0	19
gde	0	0	3	5	1	3	0	1	0	5	5	0	0	23
Derrotas0	7	30	33	4	21	34	30	22	31	31	15	9		

Tabela 149 – Contagem de melhores resultado no I_{ϵ} e W=25.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moad	sms	nsga	ssnsga	gde	Total
cross0	0	3	5	4	1	6	6	6	5	6	6	6	6	60
cross1	0	0	3	2	0	4	4	2	2	6	3	5	2	33
cross2	0	0	0	0	0	2	5	2	1	3	1	2	1	17
cross3	0	0	2	0	0	3	4	3	1	4	3	4	2	26
plasm0	0	0	3	2	0	2	5	1	2	5	4	5	0	29
plasm1	0	2	1	1	4	0	1	2	5	1	3	0	2	22
plasm2	0	0	0	0	0	0	0	0	1	0	0	0	0	1
plasm3	0	0	2	0	1	1	1	0	0	2	0	1	0	8
moad	0	0	3	2	0	0	0	0	0	0	0	0	0	5
sms	0	0	0	0	1	0	3	3	5	0	0	0	0	12
nsga	0	0	0	0	0	0	3	2	4	1	0	0	0	10
ssnsga	0	0	3	1	1	0	0	0	2	3	3	0	0	13
gde	0	1	4	2	4	1	0	1	1	4	2	0	0	20
Derrotas0	6	26	14	12	19	32	22	29	35	25	23	13		

Tabela 151 – Contagem de melhores resultado no Tempo e W=25.

	cross0	cross1	cross2	cross3	plasmid0	plasmid1	plasmid2	plasmid3	moad	sms	nsga	ssnsga	gde	Total
cross0	0	0	0	0	6	0	0	0	0	0	0	0	0	6
cross1	6	0	0	6	6	6	0	6	0	0	0	0	0	30
cross2	3	0	0	3	6	6	0	6	0	0	0	0	0	24
cross3	0	0	0	0	6	0	0	0	0	0	0	0	0	6
plasm0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
plasm1	6	0	0	6	0	0	0	3	0	0	0	0	0	15
plasm2	6	6	6	6	0	0	0	6	0	0	0	0	0	30
plasm3	0	0	0	0	6	3	0	0	0	0	0	0	0	9
moad	0	0	0	0	6	6	6	4	0	3	0	6	4	35
sms	6	0	0	6	2	4	6	6	0	0	0	0	0	30
nsga	0	3	0	0	6	6	6	3	0	4	0	6	6	40
ssnsga	6	0	6	6	0	0	4	6	0	0	0	0	0	28
gde	6	0	4	6	0	0	6	6	2	0	0	0	0	30
Derrotas39	9	16	39	44	31	28	46	2	7	0	12	10		

Tabela 152 – Contagem de melhores resultado no I_{ϵ_+} e $W=25$.

W=25												
	b30_25		b30_26		b30_27		b60_25		b60_26		b60_27	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	10	0	10	0	10	0	9	0	9	0	11	0
cross1	7	0	9	0	6	0	6	2	5	1	4	2
cross2	1	7	2	5	4	2	8	0	1	8	1	3
cross3	2	6	0	4	4	1	8	0	4	1	8	0
plasm0	6	3	6	2	4	2	5	3	2	4	3	2
plasm1	3	3	2	2	2	6	1	6	6	1	8	2
plasm2	0	7	0	6	0	7	0	5	0	5	1	5
plasm3	1	5	0	5	4	2	1	4	2	3	1	3
moead	2	4	2	4	0	4	0	5	2	5	0	6
sms	3	6	3	5	1	8	1	4	3	7	1	6
nsga	3	5	3	5	2	3	0	5	3	3	1	4
ssnsga	5	2	2	3	1	6	1	5	4	3	1	4
gde	6	1	4	2	5	2	2	3	2	2	0	3

Tabela 154 – Contagem de melhores resultado no I_H e $W=25$.

W=25													
	b30_25		b30_26		b30_27		b60_25		b60_26		b60_27		
	v	d	v	d	v	d	v	d	v	d	v	d	
cross0	11	0	10	0	9	0	11	0	10	0	10	0	
cross1	5	1	8	0	8	0	1	2	4	2	4	2	
cross2	2	6	1	6	3	4	3	2	1	7	1	5	
cross3	2	6	1	6	3	4	1	5	1	6	1	6	
plasm0	7	0	6	1	7	1	4	1	6	0	5	1	
plasm1	1	6	4	3	5	3	1	5	4	2	4	2	
plasm2	0	6	0	5	0	8	0	4	0	6	0	5	
plasm3	0	6	0	6	4	2	1	7	2	4	1	5	
moead	2	4	2	4	1	4	1	3	2	4	3	3	
sms	4	5	3	6	2	7	8	2	4	6	5	5	
nsga	3	5	3	5	1	6	2	5	3	5	3	5	
ssnsga	5	1	4	3	1	6	4	2	3	1	2	2	
gde	5	1	5	2	3	2	3	2	4	1	3	1	

Tabela 153 – Contagem de melhores resultado no I_{ϵ} e $W=25$.

W=25												
	b30_25		b30_26		b30_27		b60_25		b60_26		b60_27	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	10	0	10	0	10	0	9	0	10	0	11	0
cross1	5	0	9	0	6	0	6	2	4	2	3	2
cross2	1	7	2	6	4	2	8	0	1	8	1	3
cross3	2	6	0	4	4	2	8	0	4	2	8	0
plasm0	6	3	6	2	5	1	5	3	4	1	3	2
plasm1	3	3	2	2	2	6	1	5	7	1	7	2
plasm2	0	6	0	6	0	7	0	5	0	4	1	4
plasm3	1	5	0	5	4	2	1	4	1	3	1	3
moead	2	4	2	4	0	5	0	5	1	4	0	7
sms	3	6	3	5	1	8	1	4	3	6	1	6
nsga	3	5	3	5	2	3	0	5	1	4	1	3
ssnsga	5	1	3	3	1	6	0	5	3	4	1	4
gde	6	1	4	2	5	2	2	3	2	2	1	3

Tabela 155 – Contagem de melhores resultado no Tempo e $W=25$.

W=25												
	b30_25		b30_26		b30_27		b60_25		b60_26		b60_27	
	v	d	v	d	v	d	v	d	v	d	v	d
cross0	1	7	1	6	1	6	1	6	1	7	1	7
cross1	5	1	5	1	5	1	5	2	5	2	5	2
cross2	5	3	4	3	4	3	3	2	4	2	4	3
cross3	1	7	1	7	1	7	1	6	1	6	1	6
plasm0	0	7	0	7	0	7	0	7	0	8	0	8
plasm1	2	6	2	6	2	6	3	5	3	4	3	4
plasm2	5	5	5	5	5	5	5	4	5	4	5	5
plasm3	2	7	2	8	2	7	1	8	1	8	1	8
moead	6	0	7	0	6	0	6	0	5	1	5	1
sms	5	1	5	2	5	1	5	1	5	1	5	1
nsga	6	0	7	0	6	0	7	0	7	0	7	0
ssnsga	5	2	5	2	5	2	4	2	4	2	5	2
gde	5	2	5	2	5	2	4	2	5	1	6	1

Referências

ADAMS, A.; NICHOLAS, J.; SIADAK, W. Protocol independent multicast - dense mode (pim-dm): Protocol specification (revised). 01 2005. Disponível em: <<https://tools.ietf.org/html/rfc3973>>.

ALMEIDA, C. P. de et al. Transgenetic algorithms for the multi-objective quadratic assignment problem. In: *2014 Brazilian Conference on Intelligent Systems*. IEEE, 2014. Disponível em: <<https://doi.org/10.1109/bracis.2014.63>>.

ANDRADE, R. C.; GOLDBARG, M. C.; GOLDBARG, E. F. G. Roteamento multicast com múltiplas sessões sujeito a limite de or(c)amento. *Simpósio Brasileiro de Pesquisa Operacional*, 2015.

AURENHAMMER, F. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, Association for Computing Machinery (ACM), v. 23, n. 3, p. 345–405, sep 1991. Disponível em: <<https://doi.org/10.1145/116873.116880>>.

BATES, T. et al. Multiprotocol extensions for bgp-4. 01 2007. Disponível em: <<https://tools.ietf.org/html/rfc4760>>.

BEHE, M. *Darwin's Black Box: The Biochemical Challenge to Evolution*. Free Press, 1996. (Touchstone book). ISBN 9780684827544. Disponível em: <<https://books.google.com.br/books?id=7L8mkq4jG6EC>>.

BEUME, N.; NAUJOKS, B.; EMMERICH, M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, Elsevier BV, v. 181, n. 3, p. 1653–1669, sep 2007. Disponível em: <<https://doi.org/10.1016/j.ejor.2006.08.008>>.

CÁCERES, L. P.; LÓPEZ-IBÁÑEZ, M.; STÜTZLE, T. An analysis of parameters of irace. In: _____. *Evolutionary Computation in Combinatorial Optimisation: 14th European Conference, EvoCOP 2014, Granada, Spain, April 23-25, 2014, Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 37–48. ISBN 978-3-662-44320-0. Disponível em: <http://dx.doi.org/10.1007/978-3-662-44320-0_4>.

CHEN, S.; GUNLUK, O.; YENER, B. The multicast packing problem. *IEEE/ACM TRANSACTIONS ON NETWORKING*, v. 8, p. 311–318, 2000.

CHEN, Y.-R. et al. On multi-stream multi-source multicast routing. *Computer Networks*, v. 57, n. 15, p. 2916 – 2930, 2013. ISSN 1389-1286.

CHOPRA, S.; TSAI, C.-Y. Polyhedral approaches for the steiner tree problem on graphs. In: CHENG, X.; DU, D.-Z. (Ed.). *Steiner Trees in Industry*. Springer US, 2001, (Combinatorial Optimization, v. 11). p. 175–201. ISBN 978-1-4613-7963-8. Disponível em: <http://dx.doi.org/10.1007/978-1-4613-0255-1_5>.

CRICHIGNO, J.; BARÁN, B. Multiobjective multicast routing algorithm. In: SOUZA, J. de; DINI, P.; LORENZ, P. (Ed.). *Telecommunications and Networking - ICT 2004*. Springer Berlin / Heidelberg, 2004, (Lecture Notes in

Computer Science, v. 3124). p. 1029–1034. ISBN 978-3-540-22571-3. Disponível em: http://dx.doi.org/10.1007/978-3-540-27824-5_134.

CRICHIGNO, J.; BARÁN, B. Multiobjective multicast routing algorithm for traffic engineering. In: *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*. [S.l.: s.n.], 2004. p. 301–306. ISSN 1095-2055.

CUI, X.; LIN, C.; WEI, Y. A multiobjective model for qos multicast routing based on genetic algorithm. In: *Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing*. Washington, DC, USA: IEEE Computer Society, 2003. (ICCNMC '03), p. 49–. ISBN 0-7695-2033-2. Disponível em: <http://dl.acm.org/citation.cfm?id=950787.950985>.

DEB, K. Introduction to evolutionary multiobjective optimization. In: BRANKE, J. et al. (Ed.). *Multiobjective Optimization*. Springer Berlin-Heidelberg, 2008, (Lecture Notes in Computer Science, v. 5252). p. 59–96. 10.1007/978-3-540-88908-3. Disponível em: http://dx.doi.org/10.1007/978-3-540-88908-3_3.

DEB, K. et al. A fast elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, v. 6, p. 182–197, 2002.

DIJKSTRA, E. A note on two problems in connexion with graphs. *Numerische Mathematik*, Springer-Verlag, v. 1, n. 1, p. 269–271, 1959. ISSN 0029-599X. Disponível em: <http://dx.doi.org/10.1007/BF01386390>.

DORIGO, M.; BLUM, C. Ant colony optimization theory: A survey. *Theoretical computer science*, Elsevier, v. 344, n. 2, p. 243–278, 2005.

DORIGO, M.; STÜTZLE, T. Ant colony optimization: overview and recent advances. In: *Handbook of metaheuristics*. [S.l.]: Springer, 2010. p. 227–263.

EMMERICH, M.; BEUME, N.; NAUJOKS, B. An EMO algorithm using the hypervolume measure as selection criterion. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005. p. 62–76. Disponível em: https://doi.org/10.1007/978-3-540-31880-4_5.

FABREGAT, R. et al. Multi-objective optimization scheme for multicast flows: A survey, a model and a moea solution. In: *Proceedings of the 3rd International IFIP/ACM Latin American Conference on Networking*. New York, NY, USA: ACM, 2005. (LANC '05), p. 73–86. ISBN 1-59593-008-6. Disponível em: <http://doi.acm.org/10.1145/1168117.1168128>.

FABREGAT, R. et al. *Multitree Routing for Multicast Flows: A Genetic Algorithm Approach*. CCIA. 2004.

FENNER, B. et al. Protocol independent multicast - sparse mode (pim-sm). 03 2016. Disponível em: <https://tools.ietf.org/html/rfc7761>.

FONSECA, C.; FLEMING, P. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: *Proceedings of the 5th International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. p. 416–423.

- FOULDS, L. A multi-commodity flow network design problem. *Transportation Research Part B: Methodological*, v. 15, n. 4, p. 273 – 283, 1981. ISSN 0191-2615. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0191261581900138>>.
- FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, Informa UK Limited, v. 32, n. 200, p. 675–701, dec 1937. Disponível em: <<https://doi.org/10.1080/01621459.1937.10503522>>.
- FULKERSON, D. R. Blocking and anti-blocking pairs of polyhedra. *Mathematical Programming*, v. 1, n. 1, p. 168–194, 1971. ISSN 1436-4646. Disponível em: <<http://dx.doi.org/10.1007/BF01584085>>.
- GAMMA, E. et al. *Design Patterns - Elements of Reusable Object-Oriented Software*. [S.l.]: Pearson Education, 1994. (Addison-Wesley Professional Computing Series). ISBN 9780321700698.
- GAREY, M. R.; JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979. ISBN 0716710447.
- GIL, A. C. *Como Elaborar Projetos de Pesquisa*. [S.l.]: Atlas, 2010. 200 p.
- GOLDBARG, M. C.; GOLDBARG, E. F. G. Transgenética computacional: uma aplicação ao problema quadrático de alocação. *Pesquisa Operacional*, FapUNIFESP (SciELO), v. 22, n. 3, p. 359–386, jul 2002. Disponível em: <<https://doi.org/10.1590/s0101-74382002000300005>>.
- GOLDBARG, M. C.; LUNA, H. P. [S.l.]: Elsevier, 2005.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675.
- GUO, D. et al. Source selection problem in multi-source multi-destination multicasting. *Computer Networks*, v. 127, p. 43 – 55, 2017. ISSN 1389-1286.
- HANSEN, P.; JAZSKIIEWICS, A. *Evaluating the quality of approximations of the non-dominated set*. [S.l.], 1998.
- HEDRICK, C. Routing information protocol. 01 2017. Disponível em: <<https://tools.ietf.org/html/rfc1058>>.
- HOLBROOK, H.; CAIN, B. Source-specific multicast for ip. Agosto 2006. Disponível em: <<https://tools.ietf.org/html/rfc4607>>.
- HUANG, J.; LIU, Y. Moeaq: A qos-aware multicast routing algorithm for {MANET}. *Expert Systems with Applications*, v. 37, n. 2, p. 1391 – 1399, 2010. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417409006526>>.
- HWANG, F.; RICHARDS, D. S.; WINTER, P. *Steiner Tree Problem*. [S.l.]: Elsevier Science Publishers, 1992.

IETF, I. E. T. F. Iana guidelines for ipv4 multicast address assignments. 01 2017. Disponível em: <<https://tools.ietf.org/html/rfc5771>>.

IETF, I. E. T. F. Using internet group management protocol version 3 (igmpv3) and multicast listener discovery protocol version 2 (mldv2) for source-specific multicast. 01 2017. Disponível em: <<https://tools.ietf.org/html/rfc5771>>.

JASZKIEWICZ, A. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment. *Trans. Evol. Comp.*, IEEE Press, Piscataway, NJ, USA, v. 6, n. 4, p. 402–412, ago. 2002. ISSN 1089-778X. Disponível em: <<http://dx.doi.org/10.1109/TEVC.2002.802873>>.

JIA, X.; WANG, L. A group multicast routing algorithm by using multiple minimum steiner trees. *Computer Communications*, v. 20, n. 9, p. 750 – 758, 1997. ISSN 0140-3664. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140366497000819>>.

KANG, J.; PARK, K.; PARK, S. Optimal multicast route packing. *European Journal of Operational Research*, v. 196, p. 351–359, 2009.

KIMURA, M. Evolutionary rate at the molecular level. *Nature*, Springer Nature, v. 217, n. 5129, p. 624–626, feb 1968. Disponível em: <<https://doi.org/10.1038/217624a0>>.

KNOWLES, J.; THIELE, L.; ZITZLER, E. *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. [S.l.], 2006.

KNOWLES, J. D. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. Tese (Doutorado) — Department of Computer Science, University of Reading, UK, January 2002.

KOCK, T.; MARTIN, A. Solving steiner tree problem in graphs to optimality. v. 32, p. 207–232, 1998.

KOMPELLA, V.; PASQUALE, J.; POLYZOS, G. Multicast routing for multimedia communication. *Networking, IEEE/ACM Transactions on*, v. 1, n. 3, p. 286 –292, jun 1993. ISSN 1063-6692.

KOU, L.; MARKOWSKY, G.; BERMAN, L. A fast algorithm for steiner trees. *Acta Informatica*, Springer Berlin / Heidelberg, v. 15, p. 141–145, 1981. ISSN 0001-5903. 10.1007/BF00288961.

KRUSKAL, J. B. On the shortest subtree of a graph and the traveling salesman problem. p. 48–50, February 1956.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, Taylor e Francis, v. 47, n. 260, p. 583–621, 1952.

KUKKONEN, S.; LAMPINEN, J. Comparison of generalized differential evolution to other multi-objective evolutionary algorithms. In: *European Congress on Computational Methods in Applied Sciences and Engineering*. [S.l.]: University of Jyväskylä, Department of Mathematical Information Technology, 2004.

- KUKKONEN, S.; LAMPINEN, J. An extension of generalized differential evolution for multi-objective optimization with constraints. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004. p. 752–761. Disponível em: <https://doi.org/10.1007/978-3-540-30217-9_76>.
- KUKKONEN, S.; LAMPINEN, J. GDE3: The third evolution step of generalized differential evolution. In: *2005 IEEE Congress on Evolutionary Computation*. IEEE, 2005. Disponível em: <<https://doi.org/10.1109/cec.2005.1554717>>.
- LEE, C. Y.; CHO, H. K. Multiple multicast tree allocation in ip network. *Comput. Oper. Res.*, Elsevier Science Ltd., Oxford, UK, UK, v. 31, p. 1115–1133, June 2004. ISSN 0305-0548.
- LÓPEZ-IBÁÑEZ, M. et al. *The irace package, Iterated Race for Automatic Algorithm Configuration*. [S.l.], 2011. Disponível em: <<http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2011-004.pdf>>.
- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: _____. *Handbook of Metaheuristics*. Boston, MA: Springer US, 2003. p. 320–353. ISBN 978-0-306-48056-0. Disponível em: <http://dx.doi.org/10.1007/0-306-48056-5_11>.
- LOW, C. P.; WANG, N. An efficient algorithm for group multicast routing with bandwidth reservation. In: *Proceedings of the 7th IEEE International Conference on Networks*. Washington, DC, USA: IEEE Computer Society, 1999. (ICON '99), p. 43–. ISBN 0-7695-0243-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=850938.852543>>.
- LUKE, S. *Essentials of Metaheuristics*. [S.l.]: Lulu, 2009.
- MA, Q.; STEENKISTE, P. Quality-of-service routing for traffic with performance guarantees. In: _____. *Building QoS into Distributed Systems: IFIP TC6 WG6.1 Fifth International Workshop on Quality of Service (IWQOS '97), 21–23 May 1997, New York, USA*. Boston, MA: Springer US, 1997. p. 115–126. ISBN 978-0-387-35170-4. Disponível em: <http://dx.doi.org/10.1007/978-0-387-35170-4_12>.
- MAIA, S. M. D. M.; GOLDBARG, E. F. G.; GOLDBARG, M. C. Evolutionary algorithms for the bi-objective adjacent only quadratic spanning tree. *Int. J. Innov. Comput. Appl.*, Inderscience Publishers, Inderscience Publishers, Geneva, SWITZERLAND, v. 6, n. 2, p. 63–72, dez. 2014. ISSN 1751-648X. Disponível em: <<http://dx.doi.org/10.1504/IJICA.2014.066493>>.
- MARGULIS, L.; FESTER, R. *Symbiosis as a Source of Evolutionary Innovation: Speciation and Morphogenesis*. MIT Press, 1991. ISBN 9780262132695. Disponível em: <<https://books.google.com.br/books?id=3sKzeiHUIUQC>>.
- MARTINS, S. L. et al. Greedy randomized adaptive search procedures for the steiner problem in graphs. In: *QUADRATIC ASSIGNMENT AND RELATED PROBLEMS, VOLUME 16 OF DIMACS SERIES ON DISCRETE MATHEMATICS AND THEORETICAL COMPUTER SCIENCE*. [S.l.]: American Mathematical Society, 1999. p. 237–261.
- MASIP-BRUIN, X. et al. Research challenges in qos routing. *Computer Communications*, v. 29, n. 5, p. 563 – 581, 2006. ISSN 0140-3664. Networks of Excellence. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140366405002100>>.

MCBRIDE, M.; MEYLOR, J.; MEYER, D. Multicast source discovery protocol (msdp) deployment scenarios. Agosto 2006. Disponível em: <<https://tools.ietf.org/html/rfc4611>>.

MEDINA, A. et al. Brite: An approach to universal topology generation. In: *Proceedings of the Ninth International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. Washington, DC, USA: IEEE Computer Society, 2001. (MASCOTS '01), p. 346-. Disponível em: <<http://dl.acm.org/citation.cfm?id=882459.882563>>.

MIETTINEN, K. *Nonlinear Multiobjective Optimization*. Springer US, 1998. Disponível em: <<https://doi.org/10.1007/978-1-4615-5563-6>>.

NEMENYI, P. *Distribution-free Multiple Comparisons*. [s.n.], 1963. Disponível em: <<https://books.google.com.br/books?id=nhDMtgAACAAJ>>.

NING, J.; ZHANG, C.; ZHANG, B. A novel artificial bee colony algorithm for the qos based multicast route optimization problem. *Optik - International Journal for Light and Electron Optics*, v. 127, n. 5, p. 2771 – 2779, 2016. ISSN 0030-4026. Disponível em: <[//www.sciencedirect.com/science/article/pii/S0030402615018562](http://www.sciencedirect.com/science/article/pii/S0030402615018562)>.

NORONHA, C.; TOBAGI, F. Optimum routing of multicast streams. *IEEE INFOCOM*, p. 865–873, 1994.

OLIVEIRA, C. A. S.; PARDALOS, P. M. A survey of combinatorial optimization problems in multicast routing. *Comput. Oper. Res.*, Elsevier Science Ltd., Oxford, UK, UK, v. 32, p. 1953–1981, August 2005. ISSN 0305-0548. Disponível em: <<http://dl.acm.org/citation.cfm?id=1063435.1063437>>.

OPTIMIZATION, I. G. *Gurobi Optimizer Reference Manual*. 2015. Disponível em: <<http://www.gurobi.com>>.

PAQUETE, L.; STÜTZLE, T. A two-phase local search for the biobjective traveling salesman problem. In: *Proceedings of the 2nd international conference on Evolutionary multi-criterion optimization*. Berlin, Heidelberg: Springer-Verlag, 2003. (EMO'03), p. 479–493. ISBN 3-540-01869-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=1760102.1760142>>.

PATEL, M. K.; KABAT, M. R.; TRIPATHY, C. R. A hybrid aco/pso based algorithm for qos multicast routing problem. *Ain Shams Engineering Journal*, v. 5, n. 1, p. 113 – 120, 2014. ISSN 2090-4479. Disponível em: <[//www.sciencedirect.com/science/article/pii/S2090447913000695](http://www.sciencedirect.com/science/article/pii/S2090447913000695)>.

PINTO, D.; BARÁN, B.; FABREGAT, R. Multi-objective multicast routing based on ant colony optimization. In: *Proceedings of the 2005 Conference on Artificial Intelligence Research and Development*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2005. p. 363–370. ISBN 1-58603-560-6. Disponível em: <<http://dl.acm.org/citation.cfm?id=1565835.1565888>>.

POLUKHIN, A. *Boost C++ Application Development Cookbook*. Packt Publishing, 2013. ISBN 9781849514897. Disponível em: <<https://books.google.com.br/books?id=gCn0wqT4W14C>>.

- RANDACCIO, L. S.; ATZORI, L. Group multicast routing problem: A genetic algorithms based approach. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 51, p. 3989–4004, October 2007. ISSN 1389-1286. Disponível em: <<http://dl.acm.org/citation.cfm?id=1280290.1280472>>.
- REKHTER, Y.; LI, T.; HARES, S. A border gateway protocol 4 (bgp-4). 01 2006. Disponível em: <<https://tools.ietf.org/html/rfc4271>>.
- RESENDE, M. G. C.; PARDALOS, M. P. *Handbook of Optimization in Telecommunications*. 1. ed. [S.l.]: Springer, 2006. Hardcover. ISBN 0387306625.
- RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures. In: *Handbook of Metaheuristics*. [S.l.]: Kluwer Academic Publishers, 2003. p. 219–249.
- ROCHA, D. A. M.; GOLDBARG, E. F. G.; GOLDBARG, M. C. A new evolutionary algorithm for the bi-objective minimum spanning tree. In: *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*. IEEE, 2007. Disponível em: <<https://doi.org/10.1109/isda.2007.24>>.
- ROY, A.; BANERJEE, N.; DAS, S. K. An efficient multi-objective qos-routing algorithm for wireless multicasting. In: *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)*. [S.l.: s.n.], 2002. v. 3, p. 1160–1164 vol.3.
- SCHAFFER, J. D. *Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)*. Tese (Doutorado) — , Nashville, TN, USA, 1984.
- SHAFFER, J. P. Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association*, Informa UK Limited, v. 81, n. 395, p. 826–831, sep 1986. Disponível em: <<https://doi.org/10.1080/01621459.1986.10478341>>.
- SINGH, H. P. et al. Voip: State of art for global connectivity—a critical review. *Journal of Network and Computer Applications*, v. 37, p. 365 – 379, 2014. ISSN 1084-8045. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1084804513000647>>.
- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.*, MIT Press, Cambridge, MA, USA, v. 2, n. 3, p. 221–248, set. 1994. ISSN 1063-6560. Disponível em: <<http://dx.doi.org/10.1162/evco.1994.2.3.221>>.
- STEWART, T. et al. Real-world applications of multiobjective optimization. In: _____. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 285–327. ISBN 978-3-540-88908-3. Disponível em: <http://dx.doi.org/10.1007/978-3-540-88908-3_11>.
- STORN, R.; PRICE, K. *Journal of Global Optimization*, Springer Nature, v. 11, n. 4, p. 341–359, 1997. Disponível em: <<https://doi.org/10.1023/a:1008202821328>>.
- SUN, B. et al. Multiple constraints qos multicast routing optimization algorithm in {MANET} based on {GA}. *Progress in Natural Science*, v. 18, n. 3, p. 331 – 336, 2008. ISSN 1002-0071. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1002007107000627>>.

TAKAHASHI, H.; MATSUYAMA, A. An approximate solution for the steiner problem in graphs. *Math Japonica*, Elsevier, v. 24, n. 6, p. 573–577, 1980.

VOSS, S. The steiner tree problem with hop constraints. *Annals of Operations Research*, v. 86, n. 0, p. 321–345, 1999. ISSN 1572-9338. Disponível em: <<http://dx.doi.org/10.1023/A:1018967121276>>.

WAITZMAN, D. Distance vector multicast routing protocol. 01 2017. Disponível em: <<http://www.rfc-base.org/txt/rfc-1075.txt>>.

WANG, B.; HOU, J. C. Multicast routing and its qos extension: problems, algorithms, and protocols. *IEEE Network*, v. 14, n. 1, p. 22–36, Jan 2000. ISSN 0890-8044.

WANG, C.-F.; LIANG, C.-T.; JAN, R.-H. Heuristic algorithms for packing of multiple-group multicasting. *Comput. Oper. Res.*, Elsevier Science Ltd., Oxford, UK, UK, v. 29, p. 905–924, June 2002. ISSN 0305-0548. Disponível em: <[http://dx.doi.org/10.1016/S0305-0548\(00\)00092-7](http://dx.doi.org/10.1016/S0305-0548(00)00092-7)>.

WANG, H.; SHI, Z.; LI, S. Multicast routing for delay variation bound using a modified ant colony algorithm. *Journal of Network and Computer Applications*, v. 32, n. 1, p. 258 – 272, 2009. ISSN 1084-8045. Disponível em: <www.sciencedirect.com/science/article/pii/S1084804508000209>.

WANG, H. et al. A tree-growth based ant colony algorithm for qos multicast routing problem. *Expert Systems with Applications*, v. 38, n. 9, p. 11787 – 11795, 2011. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417411004738>>.

WANG, X. et al. Multiple many-to-many multicast routing scheme in green multi-granularity transport networks. *Computer Networks*, v. 93, Part 1, p. 225 – 242, 2015. ISSN 1389-1286. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128615003886>>.

WAXMAN, B. M. Routing of multipoint connections. *IEEE Journal of Selected Areas in Communications*, IEEE Computer Society Press, v. 6, n. 9, p. 1617–1622, 1988.

WILLEBEEK-LEMAIR, M. H.; SHAE, Z.-Y. Distributed video conferencing systems. *Computer Communication*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 20, n. 3, p. 157–168, maio 1997. ISSN 0140-3664.

WU, Z. Performance modeling of multicast groups for multiplayer games in peer-to-peer networks. In: *Distributed Simulation and Real-Time Applications, 2005. DS-RT 2005 Proceedings. Ninth IEEE International Symposium on*. [S.l.: s.n.], 2005. p. 105 – 112.

XU, Y. *Metaheuristic Approaches for QoS Multicast Routing Problems*. Tese (Doutorado) — University of Nottingham, January 2011. Disponível em: <<http://www.asap.cs.nott.ac.uk/publications/pdf/yxx-thesis.pdf>>.

XU, Y.; QU, R. A hybrid scatter search meta-heuristic for delay-constrained multicast routing problems. *Applied Intelligence*, Springer Netherlands, p. 1–13, 2012. ISSN 0924-669X. 10.1007/s10489-010-0256-x. Disponível em: <<http://dx.doi.org/10.1007/s10489-010-0256-x>>.

- YAN-LIN, W. Based qos constrained group multicast routing for multimedia communication. In: *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on*. [S.l.: s.n.], 2010. v. 6, p. 296–299.
- YEN, J. Y. Finding the k shortest loopless paths in a network. *Management Science*, JSTOR, v. 17, n. 11, p. 712–716, 1971.
- YOUSSEF, H. et al. Qos-driven multicast tree generation using tabu search. *Computer Communications*, v. 25, n. 11-12, p. 1140 – 1149, 2002. ISSN 0140-3664. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140366402000294>>.
- YU, Y.; NI, L.; ZHENG, Y. Survey of qos multicast routing protocols in manets. In: *2011 International Conference on Computer Science and Service System (CSSS)*. [S.l.: s.n.], 2011. p. 3630–3635.
- ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, v. 11, n. 6, p. 712–731, Dec 2007. ISSN 1089-778X.
- ZHENG, X.; CHO, C.; XIA, Y. Content distribution by multiple multicast trees and intersession cooperation: Optimal algorithms and approximations. *Computer Networks*, v. 83, p. 100 – 117, 2015. ISSN 1389-1286. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128615000900>>.
- ZIMMERMANN, H. Osi reference model - the iso model of architecture for open systems interconnection. *IEEE Transactions on Communications*, v. 28, n. 4, p. 425–432, April 1980. ISSN 0090-6778.
- ZITZLER, E.; KNOWLES, J.; THIELE, L. Multiobjective optimization. In: BRANKE, J. et al. (Ed.). Berlin, Heidelberg: Springer-Verlag, 2008. cap. Quality Assessment of Pareto Set Approximations, p. 373–404. ISBN 978-3-540-88907-6. Disponível em: <http://dx.doi.org/10.1007/978-3-540-88908-3_14>.
- ZITZLER, E. et al. Why Quality Assessment Of Multiobjective Optimizers Is Difficult. In: *Genetic and Evolutionary Computation Conference (GECCO 2002)*. New York, NY, USA: Morgan Kaufmann Publishers, 2002. p. 666–674.
- ZITZLER, E.; THIELE, L. *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. 1999.
- ZITZLER, E. et al. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 2, p. 117–132, 2003.